Three Dimensional Indoor Location Tracking Viewer

Chi-Shian Yang[1], Sang-Joong Jung[2], and Wan-Young Chung[3]
[1]Department of Ubiquitous IT, Graduate School of Design & IT, Dongseo University, Busan, Korea
[e-mail: zappy1130@yahoo.com]
[2]Department of Electronics Engineering, Graduate School of Pukyong National University, Busan, Korea
[e-mail: jangels3497@gmail.com]
[3]Division of Electronics, Computer and Telecommunication Engineering, Pukyong National University, Busan, Korea [e-mail: wychung@pknu.ac.kr]
*Corresponding author: Wan-Young Chung

Received December 20, 2008; revised January 27, 2009; accepted February 5, 2009; published February 23, 2009

Abstract

In this paper we develop an indoor location tracking system and its 3D tracking monitoring viewer, viz., 3D Navigation Viewer (3DNV). We focus on the integration of an indoor location tracking system with the Virtual Reality Modeling Language (VRML), to facilitate a representation of the user's spatial information in virtual indoor environments that is synchronized with the physical location environment. The developed indoor location tracking system employs beacons as active transmitters, and a listener as a passive receiver. The distance information calculated from the difference speeds of RF and Ultrasonic signals is exploited, to determine the user's physical location. This is essential in supporting third parties like doctors and caregivers in identifying the activities and status of a particular individual via 3DNV. 3DNV serves as a unified user interface for an indoor location tracking system, showing the viewpoint and position of the target in virtual indoor environments. It was implemented using VRML, to provide an actual real time visualization of the target's spatial information

Keywords: Indoor location tracking system, 3D indoor location viewer, virtual reality modeling language (VRML)

We deeply appreciate the unknown reviewers' invaluable comments and very quick responses during the review time. This work was supported by Korea Science & Engineering Foundation through the Joint Research Program, the Korean government (Grant No. F01-2005-000-10362-0). This paper was presented in part in *ITC-CSCC 2007*, Nov. 2007.

DOI: 10.3837/tiis.2009.01.006

1. Introduction

Location-aware systems and services have been the focus of much recent interest [1][2][3]. They track and display the user's location on the handheld devices (laptop/palmtop) he/she is carrying, and support the implementation of diverse services. Such systems are extremely beneficial in the context of interesting applications like non-human tourist guides in museums and efficient indoor route guidance from the current location of the user to his/her destination [4].

When navigating in an unfamiliar environment people tend to rely on maps. However, normal two-dimensional (2D) maps are not always so easy to read and use. For 2D maps, the mind must first build a conceptual model of the relief before any analysis can be made. For this reason, many adults are not proficient at using maps and they might even avoid them.

Therefore, our system employs a 3D user interface, to encourage data exploration and enable relevant data-sets to be located and visualized more effectively. The aim is to provide a better user interface, to assist users in exploring and visualizing spatial information. This is due to the fact that 3D modeling is very useful in environmental monitoring and simulation, providing information based on the user's position in the physical world.

VRML [5], which is a kind of universal interchange file format integrating 3D graphics and multimedia, is capable of representing static and animated dynamic 3D and multimedia objects and interactive performance on a wide variety of computing platforms. Owing to its platform-independent definition, VRML offers a higher-level abstraction and is therefore faster and more convenient. Especially, VRML provides a variety of nodes that serve various purposes. With the combination of these nodes, not only the physical attributes and spatial position of shared objects can be represented clearly, but also the real characteristics of shared objects can be represented easily. Apparently, VRML is a good solution for all the aforementioned functionalities: virtual world representation, distribution, and rendering.

In this paper, we proposed a prototype of an indoor location-aware system on the basis of visualization of 3D indoor environments. The application that we are addressing provides users with information about their position, via the creation of an indoor activity monitoring navigation viewer.

2. Constructing Virtual Environments

The first step was the creation of the indoor environments. It was proposed to model these using VRML, which was defined by Web3D Consortium as a platform independent format, and is supported by most common 3D packages.

2.1 VRML

VRML was the first standard (ISO/IEC 14772-1:1997) designed for distribution of 3D graphics in the Web environment. This has an ASCII file format and a syntax based on structuring units called nodes [6]. The most important nodes are related to the description of geometry (regular or irregular shapes), illumination (directional, spot, point and ambient lights), materials and textures (draping and mapping of JPEG, GIP, and PNG image file formats).

Each node has three basic aspects: appearance, geometry and dynamics. The appearance node is concerned with the solid perception of the shapes, i.e. material and texture. Applying texture to the surfaces of objects is supported in two ways: 1) Simply wrapping a surface with an image, and 2) Texture mapping. The texture mapping requires a second node to be specified, which contains the image coordinates corresponding to the coordinates of the geometry. The geometry of 3D objects can be described by using predefined primitives or by sets of faces, which are represented by coordinates of points.

The combination of other nodes, i.e. sensors, routes and interpolators introduces dynamics. Sensors detect either viewer's action or viewer position of the time. The router directs the captured event to interpolators, to alter certain fields (i.e., color, position, orientation and scale).

Hence, the reception for VRML among many developers was enthusiastic. This resulted in numerous types of VRML browsers for visualizing and navigation of 3D graphics on the Web and applications.

2.2 VRML Browsers

Using VRML browsers, the user can interact with and examine the 3D scene from all directions. The appearance of the view is highly dependent on the type of browser. Different VRML browsers, like Cosmo Player, WorldView, Cortona3D Viewer (previously known as Cortona VRML Client) have been tested, each of which differs slightly in terms of the color, size, orientation and initial viewpoint. Cortona3D Viewer [7] from ParallelGraphics is recommended, as it provides some additional extensions to the standard one, and a programming interface to access other sophisticated applications. **Fig. 1** depicts one of the cells viewed from various viewpoints via Cortona3D Viewer.

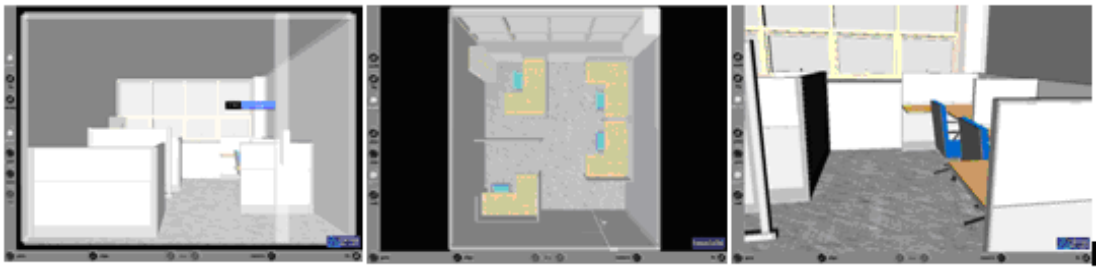

Fig. 1. A cell viewed from various viewpoints

2.3. Indoor Environments

A building of our laboratory, the Ubiquitous-IT building, Dongseo University, was the modeled building. Data provided by Computer Aided Design (CAD) only covers the basements of the buildings. The dimensions, especially objects inside rooms, had to be modeled manually, after object measuring. The need for hand-drawn buildings or manual modeling is definitely the drawback of the system. However, in the near future, the process of constructing 3D floor maps for every floor of a particular building will be automatic.

2.4. Portal Culling Algorithm

VRML gives poor performances on the visualization of large-scale areas, as it is an interpreted language rather than a compiled one. Therefore, assistant technology is necessary for better

performance. One of the most common methods is the use of Level of Detail (LOD). VRML has an embedded LOD node; however, its fast speed of transformation between different levels has the cost of a long initialization time. When a VRML file is launched, the data of all levels needs to be downloaded together to the rendering pipeline, which normally takes longer than is acceptable to users.

A better way would be to change LOD during interaction or navigation. Some work has been done on progressive rendering of VRML. In 1998 the Web3D Consortium set up a specific working group for the streaming VRML issue. However, due to the limitations of the VRML specification, progress in this area, especially in terms of its efficiency for large data sets, is still ongoing , and there has not yet been a breakthrough.

As in [8], our solution starts from the viewer's cell, considers all of its portals and checks whether they are visible or not: if a portal is visible, the cell connected to the viewer's cell by the portal will be visible as well. During navigation in the environment, it is sufficient to know when a user moves from one cell to another, and retrieve the corresponding cell data at that moment.

The culling algorithm exploits three types of visibility: cell-to-cell, cell-to-geometry and eye-to-object visibility. Cell-to-cell visibility finds which cells are visible from some point within a cell. Cell-to-object visibility is responsible for determining which objects are visible from a given cell. The third type of visibility, eye-to-object visibility, finds which parts of cells are visible from the current viewpoint.

In 3DNV, all cell management is done inside the VRML file, by a script. The script is activated when the user exits the cell he/she is currently in. At this point, the next cell is determined, all the objects in the new cell are identified, and their visibility condition is set, based on the user's estimated position and viewpoint. According to this information, the corresponding objects, which intersect the visible volume area (VVA), are loaded. The VVA of a cell is defined as the part of the user's viewpoint where the rendered objects in the cell are visible. The major advantage of this solution is that it can be used with all existing VRML browsers (both for mobile devices and desktop computers).

Fig. 2 describes the function used to cull invisible branches of nodes augmented in a tree hierarchy for cell-to-geometry visibility. An initial test is performed, to check whether the bounding box of the node (Node.BoundingBox) is entirely outside VVA. If it is not outside VVA, the node geometry is rendered (Node.Geometry) and the RenderNode is recursively called for all the child nodes (Node.Children).

```
RenderNode(Node, VVA)
  if IsBoundingBoxVisible(Node.BoundingBox, VVA)
      RenderGeometry(Node.Geometry)
      for each (Child in Node.Children)
          RenderNode(Child, VVA)
      endfor
  endif
```

Fig. 2. Pseudo code for the cell-to-geometry visibility culling and geometry rendering

Fig. 3 demonstrates the rendered scene before and after applying the culling algorithm, based on the assumption that the user is currently at the position of Cartesian coordinate 20, 25

and -15, with an orientation angle of -1.57 to the y-axis. Instead of rendering all objects in the cell that the user is currently in, after including the culling algorithm in our application, it renders only the objects within VVA.

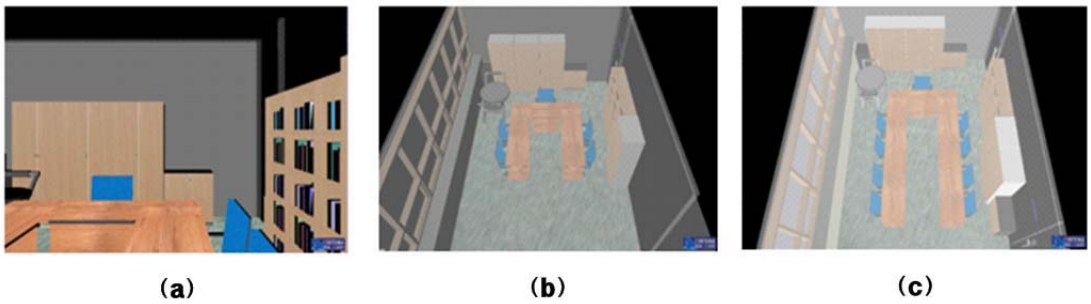

Fig. 3. Comparison between rendered scene before and after employing the culling algorithm; (a) User's current viewpoint, (b) Rendered scene without culling algorithm and (c) Rendered scene with culling algorithm

3. 3D Navigation Viewer (3DNV)

The previously developed system [9] requires Microsoft Java Virtual Machine (MSJVM) in order to manipulate 3D scenes using VRML2.0 External Authoring Interface (EAI) [2] via a Java applet. The current 3DNV utilizes the powerful VRML Automation Interface to overcome this constraint. The external environment can perform a wide range of operations on the scene using the methods and properties of VRML Automation Interface. Since we are consider the use of Cortona3D Viewer as VRML browser, VRML Automation Interface can be used as alternative to EAI.

3.1. System Architecture

The developed indoor location tracking system [10] utilizes Ultrasonic and RF transceivers to locate targets in the physical world, using Cricket wireless sensor nodes, as shown in **Fig. 4**. This passive low-cost indoor location tracking only requires the user to carry the listener during his/her navigation in the physical world. Time Difference of Arrival (TDOA) between Ultrasonic and RF signals is computed, to estimate the distance of the listener from each beacon mounted on the ceiling. The user's position is subsequently determined, as the distance information is obtained from the triangulation algorithm. 3DNV is responsible for displaying the spatial information of the target accordingly.

In 3DNV, we combine the VRML representation of the currently navigated indoor environments, which are synchronized with the physical world via the use of indoor context-aware data. The architecture of the system is depicted in **Fig. 5**. The application interface module enables the user to view the 3D world and see his status information. This module receives information from other modules and presents this to users. The information module is devoted to managing the information on the estimated position and orientation of the target. The 3D world module is required to manage the 3D representation. It exploits Cortona3D Viewer for the visualization of the 3D world and among other functionalities it must be capable of updating the viewpoint and position according to the spatial information received, and to manage large-scale environments efficiently.

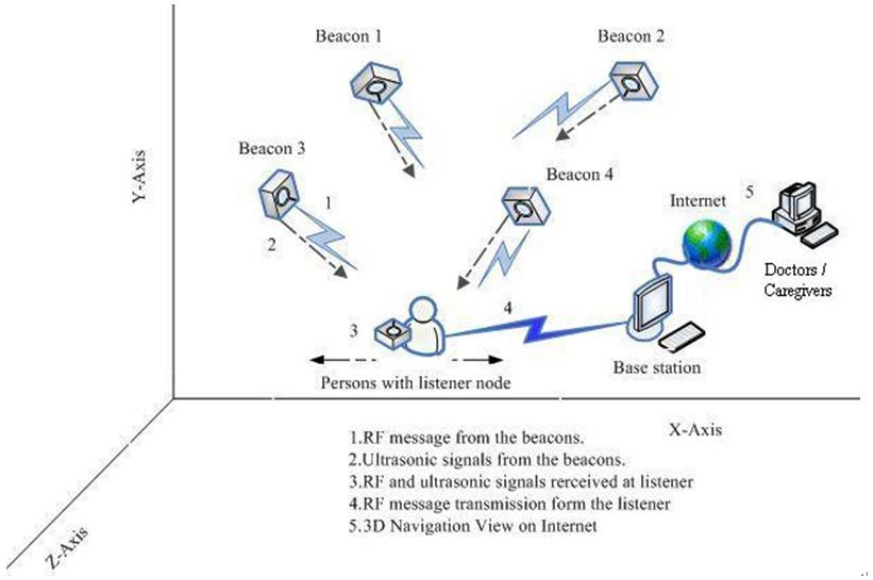

Fig. 4. Comparison between rendered scene before and after employing the culling algorithm

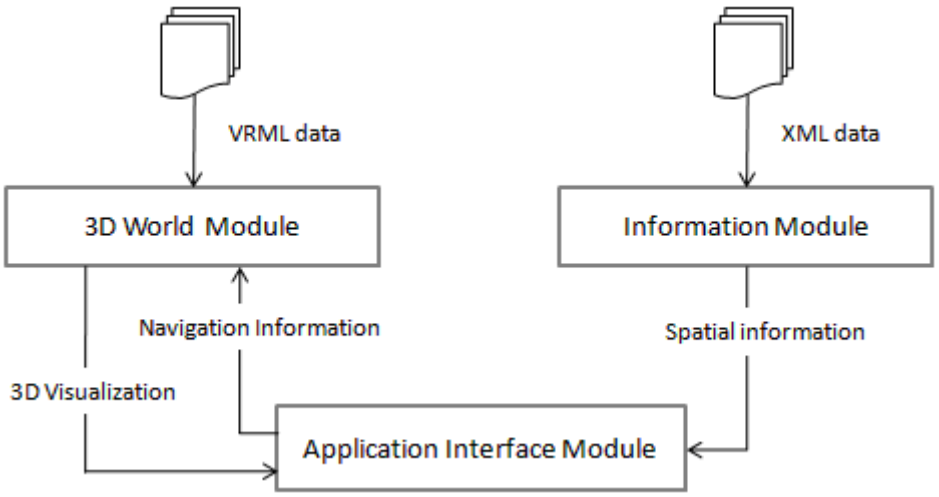

Fig. 5. Architecture of 3DNV system

3.2. Application Interface Module

Fig. 6 shows the prototype of the 3DNV system. Three main components can easily be identified. There is an upper area where the actual world is visualized based on the user's viewpoint and orientation. The lower-left area contains a map of the current indoor environment, with a marker indicating the position of the user in the virtual world. The coordinate display in the lower-right area shows the Cartesian coordinates of the current position, to help the user to find his/her position reference in the environment. In order to enable the use of ActiveX Automation Technology (ActiveX of the Cortona3D Viewer) in the .Net environment, application extension libraries are needed before-hand. These can be obtained from Cortona Software Development Kit (SDK) after installation.

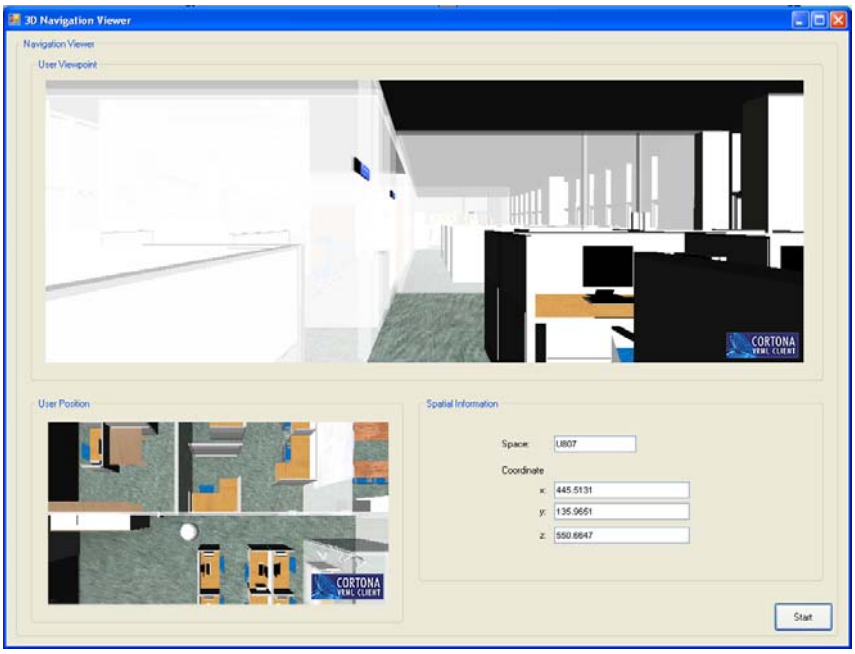

Fig. 6. 3DNV friendly user interface

3.3. Information Module

This module handles the data obtained from the indoor location tracking system, extracts the information needed, and provides this information to the application interface module. The information is stored in XML files that are formatted accordingly, with a predefined Document Type Definition (DTD). The purpose of the DTD is to define the legal building blocks of the XML documents, and to facilitate the automatic generation of XML system files from a database.

3.4. 3D World Module

The 3D world module accesses and manipulates the VRML nodes representing the 3D indoor environment shown to the user. The most important responsibility is to correctly orient the viewpoint of the 3D world, and to show the current user position in the world, according to the data provided by the indoor location tracking system via VRML Automation Interface.

Although VRML is a great solution for constructing, distributing and rendering 3D worlds, it does not support the multi-user collaboration requirements of 3DNV. An efficient technology is needed to fill the gap between a pure VRML world and 3DNV. Fortunately, VRML provides VRML Automation Interface, a set of interfaces which enables an external environment to access and manipulate various objects in a VRML scene loaded in Cortona3D Viewer.

In order to handle a large architectural model, we applied the principles of the occlusion culling algorithm. The portal culling algorithm is used to selectively remove part of the scene before it is sent through the graphics pipeline, thus reducing the number of polygons that will be processed, as described in Section 2.4.

4. System Implementation & Evaluation

The 3DNV system was implemented in C# for the desktop environment. VRML is used as the description language for the 3D world, and Cortona3D Viewer is used to display the 3D world in the application. In the current version of 3DNV, all data (i.e., geometry data on the environment and information on the objects) is stored locally.

For a practical test of the system, active beacons were deployed on ceilings according to the pre-defined local reference coordinate system. The accuracy of the coordinates was approximately 7~12cm, each node was installed at a 45 degree angle from the ceilings, and the distance between the beacons was less than 400cm. 3DNV receives the spatial information from the tracking system, and updates the 3D world accordingly. The accuracy of location mapping in the 3D world was proved to be acceptable; experiments were performed to evaluate the position indicator in the 3D environment, using the one estimated by the indoor location tracking system. An important problem was that the accuracy of the positioning was occasionally low, due to the poor precision of the received spatial information. The precision of the positional data must be improved, to provide a satisfactory level of navigation assistance to users by means of 3D representations. Another related problem concerned the viewpoint: when the user is moving, his/her current orientation can automatically be obtained from the spatial information, but when the user stops, estimating the orientation can be more difficult, if not impossible. In order to improve the orientation accuracy, the best solution is to employ an electronic compass, even if this increases the weight and size of the mobile device.

With respect to the 3D representation, the frame rate achieved was greatly influenced by the time interval between subsequent indoor tracking position data. The performance differs slightly, due to the various implementations of VRML browsers. In order to accelerate the display speed, additional 3D graphics hardware is necessary. The portal culling algorithm was incorporated, with the aim of accelerating the rendering speed and smoothly managing the complex 3D worlds by reducing the number of primitives that need to be rendered.

Table 1. Average time per frame of the system, with and without culling

Time Per Frame / Cells	Without Culling		With Culling		Speedup (%)
	Rendering Time (sec)	Document Weight (KB)	Rendering Time (sec)	Document Weight (KB)	
U801	1.7	0.86	0.6	2.01	64.7
U802	1.7	0.88	0.6	1.99	64.7
U803	1.8	0.99	0.7	2.95	61.1
U804	1.7	0.98	0.6	2.41	64.7
U805	1.7	0.91	0.6	2.12	64.7
U806	1.5	0.97	0.4	2.17	73.3
U807	1.7	1.02	0.7	2.27	58.8
U808	1.4	0.63	0.6	2.29	57.1

Assumption: Position 20 15 -15 Orientation 0 1 0 0

Table 1 reports the measured rendering time (seconds) with and without the culling algorithm described in Section 2.4. Based on this assessment, the time required is reduced by approximately 64 percent. This evaluation was performed on a Pentium 4 computer equipped with a CPU3GHz, 504MB of RAM memory and Intel 82915G /GV/910GL Express Chipset at a 1,280x1,024 resolution. The weight of documents per cell varies from a minimum of 1.99

and 0.63, with and without culling, respectively, up to a maximum of 2.95 and 1.02, with and without culling, respectively, Thus, the achieved rendering speed using this final 3D model depends on the number of visible cells and objects and on the user's spatial information.

In general, users had no difficulty matching objects in the physical world with the 3D representation. Therefore, beyond doubt, this is the best tool for demonstrating and presenting spatial information It is especially valuable for activity monitoring in particular indoor environments.

Fig. 7 illustrates the result of 3DNV updating of the user's position and viewpoint in the 3D indoor environment, as the user navigates the physical world. As the user was moving in cell U803, his/her viewpoint, orientation and position were updated according to his/her spatial information.

5. Conclusions

The development of 3DNV systems has been dominated by the integration of indoor location tracking systems and 3D graphics technology, which is replacing the dull, hard-to-use 2D text-based interface for navigation. Presently, the 3DNV system features a manually drawn 3D floor map, an indoor location tracking system using Cricket, a portal culling algorithm for accelerating rendering, and a VRML Automation Interface for displaying the target's viewpoint and position, which is synchronized with the physical world.

References

[1] Liu Xiangqian, Zhao Gang and Ma Xiaoli, "Target localization and tracking in noisy binary sensor networks with known spatial topology," *Wireless Communications and Mobile Computing*, Early View Version, 2008.

[2] Arvind Rapaka and Sanjay Madria, "Two energy efficient algorithms for tracking objects in a sensor network," *Wireless Communications and Mobile Computing*, Vol 7, Issue 6, pp. 809-819, August 2007.

[3] Jun Xu, Xuemin (Sherman) Shen, Jon W. Mark, Jun Cai, "Mobile location estimation for DS-CDMA systems using self-organizing maps," *Wireless Communications and Mobile Computing*, Vol 7, Issue 3, pp. 285-298, March 2007.

[4] Peter Ruppel, Georg Treu, Axel Küpper, and Claudia Linnhoff-Popien, "Anonymous User Tracking for Location-Based Community Services," *LNCS*, Vol. 3987/2006, pp. 116–133, May. 2006.

[5] The Web3D Consortium, The Virtual Reality Modeling Language, ISO/IEC Std 14772-1:1997 and ISO/IEC14772-2:2004,http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97 (last retrieved in 2004).

[6] H. T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 16–20, Mar. 2003.

[7] Cortona3D Viewer, http://www.cortona3d.com/cortona (last retrieved in 2006).

[8] T. Aila and V. Miettinen, "dPVS: An Occlusion Culling System for Massive Dynamic Environments," *IEEE Computer Graphics and Application*, 24(2):86–97, 2004.

[9] C.-S. Yang and W.-Y. Chung, "Development of 3-D Viewer for Indoor Location Tracking System Using Wireless Sensor Network," *Journal of the Korean Sensors Society*, 16(2):110–114, 2007.

[10] W.-Y. Chung, V.K. Singh, D.-U. Jeong, R. Mylylae, and H. Lim, "Passive and Cost Effective People Indoor Location Tracking System for Ubiquitous Healthcare," *Wireless and Optical Communication MultiConference*, pp. 538–541, Jul. 2006.

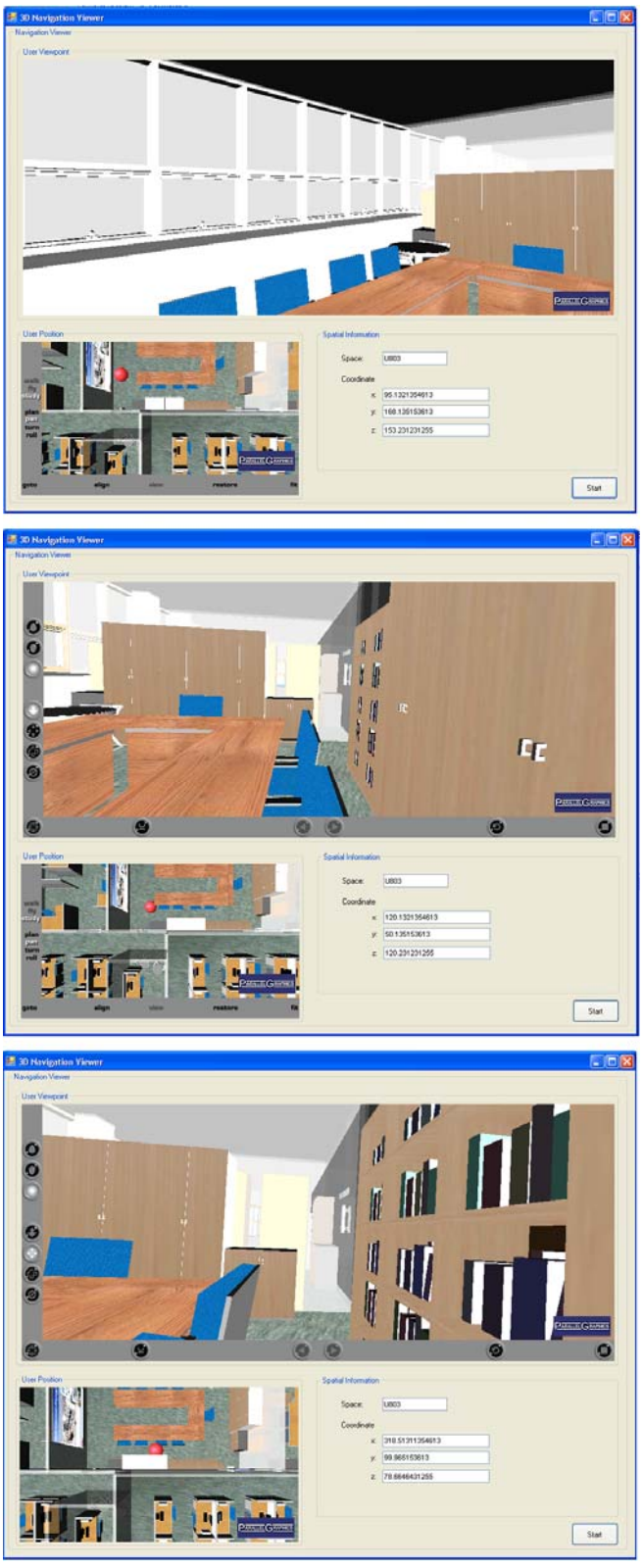

Fig. 7. 3DNV with capability to periodically specify user's viewpoint and position

Chi-Shian Yang received her M.S. degree in the Department of Ubiquitous IT from Dongseo University, Busan, Korea in 2008. She earned a Bachelor degree from MMU in Malaysia in 2005. Her areas of interest include Human-Computer Interfaces for Ubiquitous Computing, Ubiquitous Healthcare, Wireless Sensor Networks and Computer Graphics.

Sang-Joong Jung received his B.S. degree in Electronic Engineering from Dongseo University, Busan, Korea, in 2007, and his M.S in Ubiquitous IT from Dongseo University, Busan, Korea, in 2009. Since 2009, he has been a Ph.D. student at Pukyong National University, Busan, Korea. His areas of interest are Ubiquitous Healthcare, Wireless Sensor Networks, Embedded Systems and Analog Circuit Design.

Wan-Young Chung is an Associate Professor of Division of Electronics, Computer and Telecommunication Engineering at Pukyong National University, Korea from September, 2008. He earned B.S. and M.S. degrees in Electronic Engineering from Kyungpook National University, Daegu, Korea in 1987 and 1989, respectively and a Ph.D. degree in Sensor Engineering from Kyushu University, Fukuoka, Japan in 1998. From 1993 to 1999, he was an assistant professor at Semyung University. From 1999 to 2008 he was an associate professor at Dongseo University. His areas of interest include Ubiquitous Healthcare, Wireless Sensor Networks and Embedded Systems.