J. of the Korean Sensors Society
Vol. 16, No. 3 (2007) pp. 220 – 228

Location-aware visualization of VRML models in indoor location tracking system

Chi-Shian Yang and Wan-Young Chung*†

Abstract

For many applications particularly in navigation system, a three-dimensional representation improves the usability of information. This paper introduces 3D Graphical User Interface (GUI) of indoor location tracking system, 3D Navigation View. The application provides users a 3D visualization of the indoor environments they are exploring, synchronized with the physical world through spatial information obtained from indoor location tracking system. It adopts widely used Virtual Reality Modeling Language (VRML) to construct, represent, distribute and render 3D world of indoor environments over Internet. Java, an all-purpose programming language is integrated to comprehend spatial information received from indoor location tracking system. Both are connected through an interface called External Authoring Interface (EAI) provided by VRML. Via EAI, Java is given the authority to access and manipulate the 3D objects inside the 3D world that facilitates the indication of user's position and viewpoint in the constructed virtual indoor environments periodically.

Key Words : indoor location tracking system, three-dimensional (3D), VRML, EAI

1. Introductions

Unlike the conventional desktop computing, ubiquitous computing (Ubicomp) shifts computer that equipped with capability of computational, sensing and networking to the background to support users in their daily life. Location-aware computing is one of the applications of Ubicomp, particularly in indoor tracking[1-3]. Using location-sensors, computer responds to changes of user's location in physical environment and thereafter infers user intent to give him the most appropriate support for the task he is performing.

In most of the navigation systems, users are usually been traced and their current location is subsequently indicated and marked on a floor map. To provide such service, adopted solution usually employs graphical representation such as flat 2D maps which being familiar to user. This conservative type of presenting spatial information of users requires some experience on the side of users when interpreting the information presented. The use of 3D representation of the indoor environment may present a more intuitive and effective way to present spatial information to users even though the previous 3D modeling in the indoor tracking is very difficult to find.

The main idea for this approach is to generate a 3D view that would represent indoor environments in the scene, providing user with much better orientation about the space than in 2D form. Currently there exist several technologies that allow for the generation of 3D views on a computer screen. The most popular technique is various 3D modeling techniques based on the use of VRML[4].

In this paper, we present 3D Navigation View, a Graphical User Interface (GUI) for the developed indoor location tracking system[5]. The interface is used to provide users a 3D visualization of the physical indoor environments they are exploring, synchronized with their movement in physical world.

2. System Design

A new computing platform consists of small battery-powered motes with limited computation and radio communication capability is introduced due to the rapid development in embedded computing systems. Such sensor-based networking structure is used particularly in low-powered, constrained resources networked applica-

Department of Ubiquitous IT, Graduate School of Design & IT, Dongseo University, Busan 617-716, Korea
*Division of Computer & Information Engineering, Dongseo University, Busan 617-716, Korea
†Corresponding author: wychung@dongseo.ac.kr
(Received : February 27, 2007, Accepted : May 9, 2007)

tions such as ubiquitous healthcare applications[6,7] which permit data gathering and computation to be deeply embedded in the physical environment.

Through the embedded sensors, Ubicomp systems which characterized as an element of context-awareness, perceive the user's current context, in order to best meet his needs in a particular situation. This is called context-awareness or location-awareness if only the user's context is used by the application to adapt to his needs.

New technologies for the 3D scene description, such as VRML, open new possibilities to visualize large amounts of information in a new way. These technologies used through the Internet facilitate the novel approach of integration with location-aware applications.

2.1. Requirements and Analysis

Constructing the 3D indoor environments was the first step of work which can split into three distinct stages.

The first stage is getting data for the buildings and indoor environment to be constructed. The efficiency of 3D world generation is greatly dependent on the data required by the next stage. Construction with complex procedure will definitely complicate the process of the task as more data is required. Furthermore, locations of the objects inside the particular building within their 3D environment are necessary. The manual input required to produce the desirable data relies on the level of details required in the 3D world.

To construct indoor environment from the data gathered at the second stage, floors as well as the objects inside a building are modeled and placed in a 3D world. Although the primary aim is to construct indoor environments, however, the amount of details required in the environments need to be carefully considered. The more detail that is required in the constructed environments not only affects the complexity of the construction process but will greatly complicate the process to obtain the data required.

The final part is to export data in a format that can be rendered and thus render the data. Every object inside the buildings must be translated to their accurate position in the modeled environment to generate the 3D world. This requires additional input data to shift an object inside a particular building to its location. For complicated worlds, each building and object will requires its own parameters to describe their correct position in the world, this makes the amount of input data required to be greater and hence further complicate the task of generating it.

2.2. Design for Acquiring Data

The two methods identified to acquire the input data for the construction are to extract data from two-dimensional building plans and manual generation of input data.

Of the two options evaluated to acquire the input data, manual creation of input data was chosen. Automatically extracting data from plans would greatly reduce the effort required to generate input data for the construction process if achieved accurately and robustly. However, from previous work[8] it can be seen that this was a complex procedure which has not been achieved satisfactory. Due to the additional world required, it might also reduce the scope of the project as much of the time would be spend performing this task. So this automated approach has not been chosen as the small saving in manual input does not balance the time required to produce a techniques which is so dependent on the structure of the input data.

Although the chosen method of generating the input data required manual input, it should be simple and quick to produce for basic structures. This method will also be robust as it doesn't rely on poorly structured and error prone input data. As the input is simple and well structured an error on input can be easily corrected. Creating the data required to locate objects in a 3D world would require manual input no matter what method was chosen as it is unlikely a single 2D plan would contain all the information required to construct a 3D world. Thus, choosing manual creation of input data will not make this task require additional manual input.

2.3. Design for Constructing a 3D World

3D visualization should be used to improve the exploration quality, VRML is a suitable technological basis, because it is designed for the use in world wide web (WWW) and it can be easily combined with Java.

The 3D indoor environment for the indoor location tracking system includes 3D models of the real-world objects. In addition to the models of these "physical" objects we also included geometric representations of the abstract data in the scene by using real-world metaphors. They are modeled using VRML as preferred.

2.4. Design for Visualizing the 3D World

The rendering of indoor environment involves interpreting the data and constructing a 2D image to represent a 3D model using shading algorithm. Different construction methods require different rendering methods. The result from the construction stage can be manipulated for rendering.

Rendering can be performed with a simple program or can be read into 3D graphics software. The simple graphics program will display the 3D world so that it can be explored and it features clearly displayed. As the chosen method of construction is to use VRML, thus VRML plug-in or browser[10] was chosen as the interface to use with Java to render the VRML constructed 3D world.

3. 3D Modeling By VRML

Virtual reality is an emerging technology that promises significant advances in the area of user interfaces applicable to various application domains. It is a powerful means to increase the cognitive effectiveness of virtual indoor environments, virtual buildings or virtual campuses for naive end users. Recent expansion of such technology has been possible due to introduction of VRML.

How do you create an application that is low cost to end user, presented over the Internet that can be compatibly and reliably viewed and operated by virtually? We preferred VRML for the interface. The only caveat was the end users had to download and install one particular type of free VRML Internet Explorer plug-in or browser to view the 3D aspects.

3.1. VRML Standardization

Today version of VRML is very powerful, but VRML alone is insufficient. There is a need to use the ability of other language, which can communicate with VRML and which can exchange the data between. The combination of language used today are VRML-JavaScript, VRML-Java, etc. This combination is required as VRML does not support multi-user access to its virtual world, and it does not support streaming data which need to use in multi-user world.

3.2. VRML Specifications

VRML is an open, extensible, industry-standard scene description language for 3D scenes on the Internet. VRML supports lights, shadows, collision detection, object interaction, avatar creation and 6 modes for navigation through the world. It describes 3D models in form of nestable "nodes". Nodes generally define 3D physical descriptions that may be made up of 3D primitives, such as spheres, cuboids, cones and cylinders, or of complex polyhedron. In addition to these form descriptions, nodes can also define materials, colors, texture maps, lighting, shape transformations and viewing criteria.

It not only is possible to program the behaviors of three-dimensional shapes, it is also possible to create entirely new relationships to three-dimensional objects while conforming to the VRML2.0 specification. These relationship must be defined in the nodes. For instance, Fig. 1 illustrates a file which consists of nodes which create a VRML2.0 world with a rectangular table at its center. As nodes have default values, every aspect of a

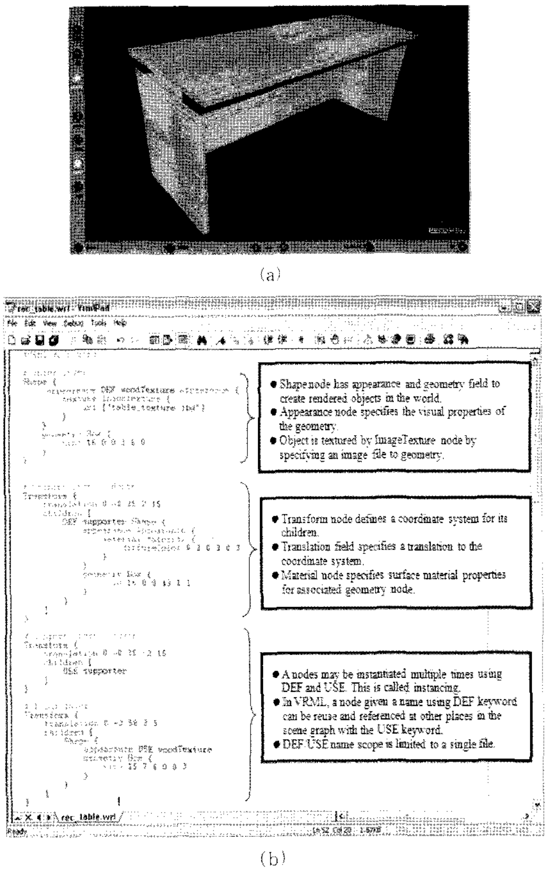


Fig. 1. An example of how a 3D object is constructed using VRML; (a) VRML world which contains a rectangular table, (b) Source code to model the rectangular table.

given node may left unspecified.

From the Fig. 1(b), "appearance" is the filed name. Each node has qualities that can be manipulated, called fields. "Appearance" is the name of the node associated with that field. The Shape node is the primary node in this scene. The appearance field specifies the look of a Shape node. The material field specifies the material that can make up the appearance, such as JPEG image for texture mapping. This is one of the example of a VRML file.

In our implementation, we consider the building of our laboratory, Ubiquitous-IT building, Dongseo University as the modeled building. Data provided by CAD as shown in Fig. 2(a), only the basements of the building, imperative dimensions especially the walls and objects inside rooms had to be measure manually. The VRML-constructed rooms of the building are presented in Fig. 2(b)-(d).

Fig. 3 highlights quality of the constructed 3D indoor environment to ensure the realism of the world and the features of the environment are accurately modeled. To convey the naturalness of the real world, image texture was used to improve the realistic of indoor environment. Although it was drawn manually, we believe that in near future the process of modeling would be done automatically.

3.3. VRML Plug-ins / Viewers / Browsers

VRML files can be viewed using a variety of plug-ins freely available on the market[9], such as Cortona VRML Client, SGI CosmoPlayer, WorldView, etc. Aside from

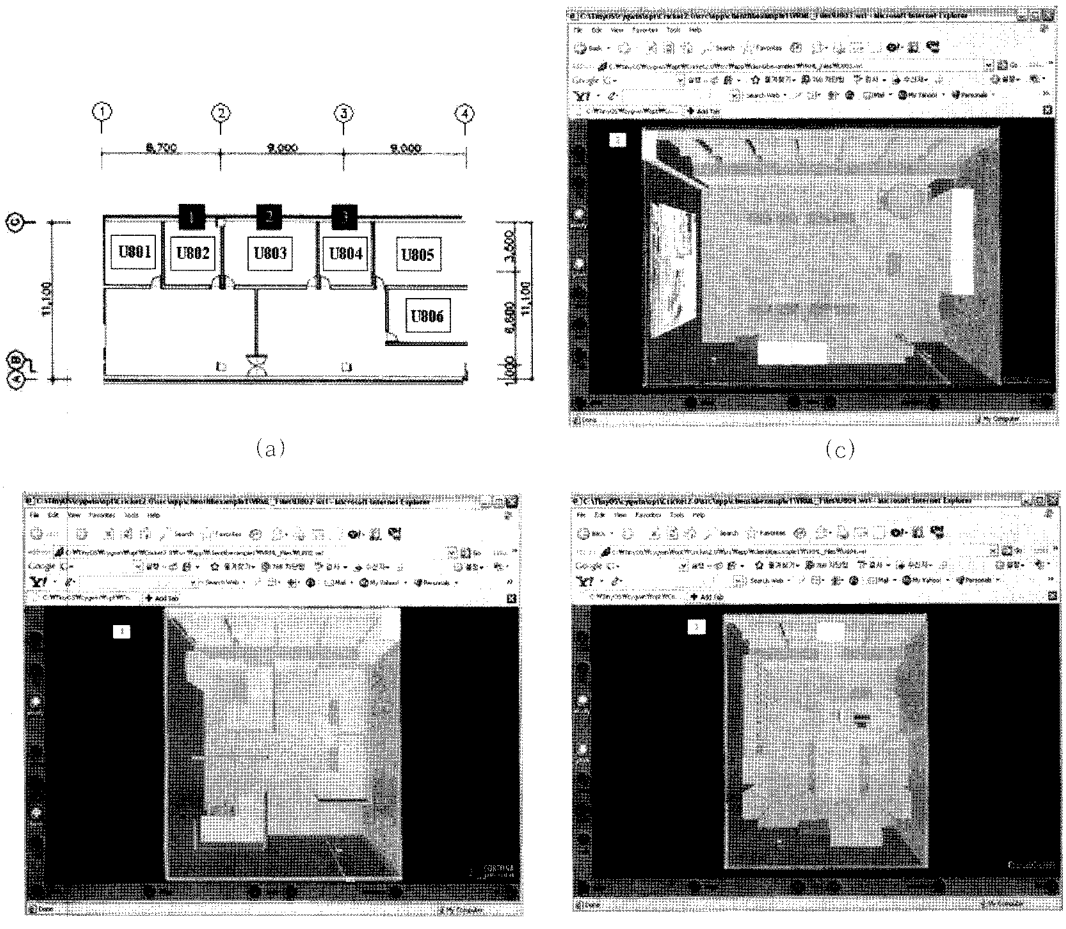
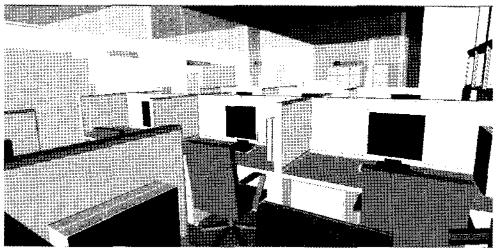


Fig. 2. 3D indoor environment modeling; (a) CAD floor map of modeled building, 8th floor U-IT building of Dongseo University, (b)-(d) The constructed indoor environment for each room.

(a)



(b)

Fig. 3. The realistic of VRML constructed indoor environment compared to the physical environment; (a) The actual environment of the room, U808, (b) The virtual environment of the room, constructed via VRML.

the nature of plug-ins, there are several other inconsistencies over platforms - the quality of Java Virtual Machine (JVM) as well as the feature sets of each browser. All of the work of our implementation was done in Internet Explorer 6.0 with the Cortona VRML Client plug-in[10] on Microsoft Window XP Professional version 2002. Use Microsoft Virtual Machine (MVM) instead of JVM or the Java applet will not work.

3.4. Constraints of VRML

As mentioned before, VRML is inherently programmable and extensible. As VRML does not aim to support full programming functionalities, it is required to work with other languages to carry out extra functionalities. The current languages to use with VRML are VRMLScript, JavaScript and Java. When programming in Java, it is possible to access the class file from within the scene itself or from outside the scene through External Authoring Interface (EAI)[4], the method which is proposed "Annex" to the VRML2.0. VRMLScript is supported by a Script Node in VRML. It is a subset of JavaScript which focuses, appropriately enough, on the activities which are most commonly done in VRML especially animation. It is also possible to use JavaScript to script events. It is supported by both major VRML browsers and is a relatively safe bet for that reason.

4. 3D Object Manipulation Via External Authoring Interface (EAI)

The VRML ushered in a new era in computer graphics by providing the first international standard 3D format for the Web. Unfortunately, in many cases, VRML applications had to be extended to include things VRML lacked, such as sophisticated user interface and interactivity, database access, multi-user support, security and system integration support. These important aspects of modern systems were added via a programming interface, called the External Authoring Interface.

The EAI was created due to the need to communicate with VRML scene from outside of that scene. By doing this, the VRML scene is no longer at the center of the architecture, but is more of a resource upon which the primary program can act and to which it can respond. This facility is used extensively in our work to update user's viewpoint and position within the VRML file through Java applet.

4.1. The Anatomy of an EAI Example

In order for Java to be able to manipulate VRML using EAI, the easiest approach is to define the node using the keyword DEF. Java applet includes EAI packages that are included with any VRML plug-in. Java applet communicates with a VRML world by first obtaining an instance of the Browser class. This class is the Java encapsulation of the VRML world. It contains the entire Browser Script Interface as well as the getNode(·) method, which returns a node given a DEF name string. Only DEF names in the VRML file are accessible.

Once the node instance is obtained from the getNode(·) method of the Browser class, its EventIns and EventOuts can be accessed. EventIns and EventOuts in VRML are objects which handle specific events for a particular node. If a node generates a particular action, an EventOut is used. If to affect a particular node, EventIn of the node must be manipulated. As shown in Fig. 4, once the node in VRML scene is engaged, translation field in the "object" node can be set through object.getEventIn ("translation"). Thus, the object is then being translated to its new location in the 3D world.

4.2. Handling of VRML scene

Through EAI, all forms of direct communication

```
import java.applet.*;
import vrml.external.*;
import vrml.external.field.*;

public class system extends Applet {

  Browser browser=null;
  Node object;

  protected static EventInSFVec3f set_translation;

  public void start() {
     browser = (Browser)vrml.external.Browser.getBrowser(this, null, 0);

     if (browser  != null) {
         // get the handle for the object
         object = browser .getNode("object");

         // get the reference to the set_translation event
         set_translation = (EventInSFVec3f) object.getEventIn("translation");
     }

     set_translation.setValue(coordinateToVRML);

  }
}
```

Fig. 4. Integration between VRML and Java through EAI.

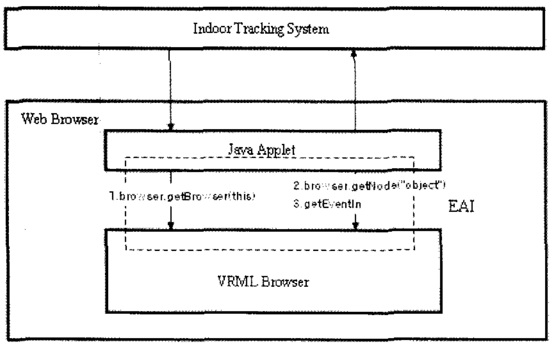


Fig. 5. Handling of VRML scene through EAI.

between VRML and scripting language like Java are possible. In our implementation, only two out of four types of accesses provided are in used: accessing the functionality of the Browser Script Interface and sending events to EventIns of nodes inside the VRML world, as depicted in Fig. 5.

User's location and viewpoint in 3D environment is periodically updated from Java applet, according to the data provided by indoor location tracking system so that 3D representation is always consistent with the actual user's view and position in the real world.

4.3. 3D Navigation View

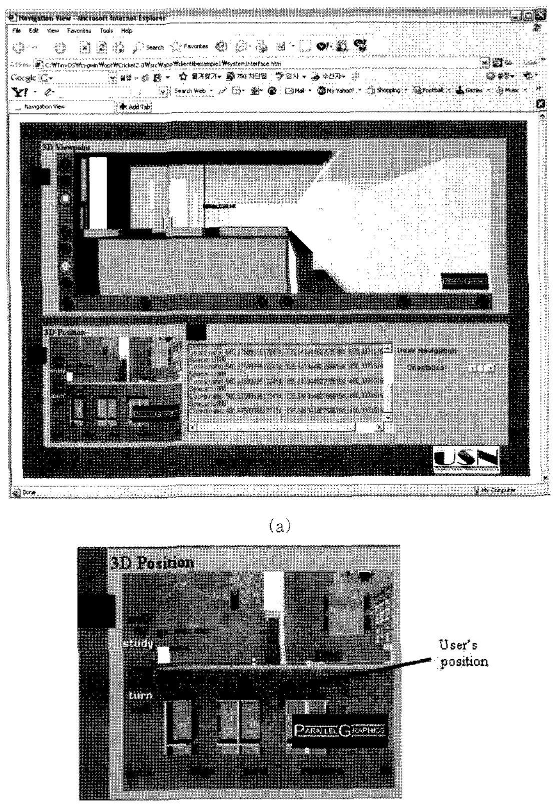
Java and VRML together offer the promise of functionality on the Web that is truly interactive and more useable than the current Web. The technology that bridges VRML and Java is still quite erratic, but with the advances in Java technology, it is possible that in a few years VRML, may be the mainstay of the Web.

VRML file is created beforehand containing all the 3D objects in the indoor environment. An additional approach to relieve navigation limitations of the VRML browser is the hybrid user interface that has been designed to support users with additional features through Java applet.

In our implementation, Java applet can be divided into two functional parts - awt part and EAI part. These two parts work on different activity upon its specific field and they exchange data between each other.

Awt part is the only visible part. Its objective is to display user interface, which to get results displayed. From the functional point of view, this part is responsible for user-computer communication using awt classes, which are included in Java in the standard installation. Dependently on the spatial information retrieved and extracted from indoor location tracking system.

EAI handles the communication between applet and VRML world located at the same Web page. This part of applet just sends processed data for 3D visualization



(a)

(b)

Fig. 6. The 3D-style interface for indoor location tracking system.

into VRML world using EAI. The main disadvantage of EAI is that the virtual world and the applet have to be on the same Web page.

Fig. 6 illustrates the 3D-style interface of the prototype system developed. Three main parts can be easily identified: an upper area where the actual 3D world is visualized with user's current viewpoint and a lower left area indicating user's location in the virtual world and a lower right area providing status information about user's current position.

We did not build a program that has to be installed permanently on client machine. Instead we developed a Java applet class which is downloaded through the Internet and executed within the Web browser on the client side. The only installation effort that appears on the client side is to download and setup a VRML plug-in, which extends the Web browser with the capability to display VRML scenes.

A combination of traditional user interfaces item (Java) and the VRML interface was used in order to provide a richer choice of interactions. The main aim was to offer users a structured and easy to manage interface, instead of overwhelming them with a huge amount of windows and options.

5. Experiments and Results

Fundamental test was carried out to observe the accuracy of user location positioning in 3D indoor environments with the aid of indoor location tracking system. In order to conduct the experiment, formerly configured beacons are mounted on the ceiling of the physical indoor environments of our laboratory. The travel path was first decided before the testing which is from U802 to U803 through U807, as shown in Fig. 7.

Apart from that, the quality of the constructed world was evaluated to ensure the realism of the world. Another concern was to test the usability of this 3D graphics whether it benefits the indoor location tracking system in terms of visualization, navigation and performance.

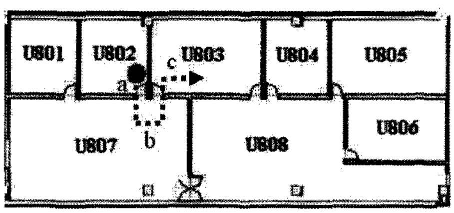


Fig. 7. Travel path set for experimental purpose.

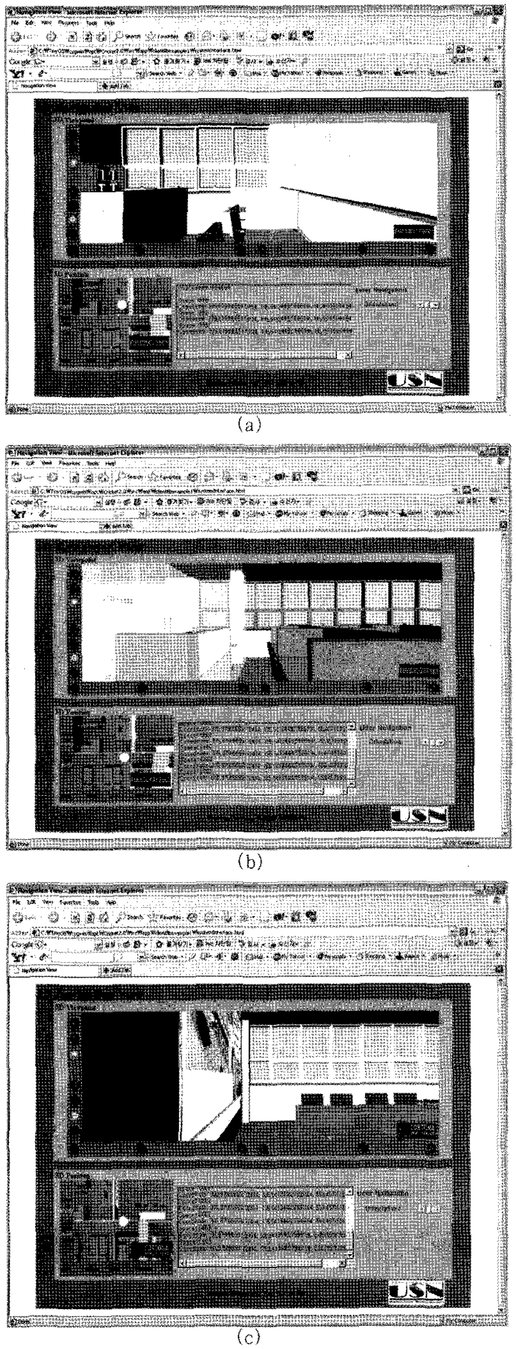


Fig. 8. 3D navigation view with the capability to update user's position and viewpoint periodically in 3D indoor environment when user travels from U802 to U803 through U807.

The performance concerning the construction and displaying of VRML scene is greatly depends on the client machine's processing power. The performance is slightly different due to the various implementation of VRML plug-ins. To accelerate the displaying and rendering speed, additional 3D graphics hardware is necessary.

According to the usability study, as predictable, users had no difficulty matching objects in the physical world with the 3D representation where 3D model is more recognizable to them than flat 2D representation of reality.

With proposed user positioning scheme, 3D Navigation View reads and extracts useful spatial information from indoor location tracking system and with the latest information, updates user's viewpoint and location in VRML worlds.

The location mapping in 3D environment is highly depends on the accuracy of location determining of indoor location tracking system. The desired accuracy can be achieved only if the spatial information provided by indoor location tracking system is sufficiently fine-grained and precise.

Fig. 8 depicts the result of 3D Navigation View updating user's position and viewpoint in the virtual 3D indoor environments as user moves freely in physical world. Java applet reads and extracts spatial information from indoor location tracking system when user travels from U802 to U803 through U807 and subsequently updates user's location and viewpoint in 3D worlds through EAI.

6. Conclusions

Throughout this paper we discussed the advantages of 3D visualization on the internet and how technologies such as Java and VRML can be employed to display user's viewpoint and location periodically over Internet.

Internet has transformed from a communication system to a global information system. This development has been initiated by the introduction of internet technologies, which provide an easy access for everybody to the variety of multimedia resources. Especially the integration of new standards in programming (Java) and 3D visualization in VRML are leading to a new type of applications for the world wide web.

These features had a vital influence on the design and implementation of the indoor location tracking system. The novel approach of aggregating it with 3D graphics technologies would be one of the solutions for the promise of providing users an interactive and appealing graphical user interface (GUI).

3D Navigation View, the developed GUI for indoor location tracking system was implemented with the aim of providing users a tremendous visualization of physical indoor environments indicating their location and viewpoint. It encapsulates Java applet to obtain spatial information and VRML worlds to display user's location and viewpoint in the 3D indoor environments.

References

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system", *ACM Transactions on Information System*, vol. 10, no. 1, pp. 91-102, 1992.

[2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF based user location and tracking system", *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, vol. 2, pp. 775-784, 2000.

[3] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications", *Proc. ACM MOBICOM Conference*, Rome, Italy, vol. 7, pp. 1-14, July 2001.

[4] The Web3D Consortium, "The virtual reality modeling language", *ISO/IEC Std 14772-1:1997 and ISO/IEC 14772-2:2004*, http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14771-VRML97/

[5] 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", *Proc. Wireless and Optical Communication MultiConference*, pp. 44-46, 2006.

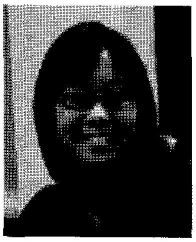
[6] 이영동, 정완영, "유비쿼터스 헬스케어를 위한 센서 네트워크 기반의 심전도 및 체온 측정 시스템-1. 센서 네트워크 플랫폼 구축", 센서학회지, 제15권, 제5호, pp. 362-370, 2006.

[7] 이대석, 정완영, "유비쿼터스 헬스케어를 위한 센서 네트워크 기반의 심전도 및 체온 측정 시스템-2. 생체신호 모니터링 소프트웨어 시스템", 센서학회지, 제15권, 제6호, pp. 417-424, 2006.

[8] T. Drury, "Generating a three-dimensional campus model", 2001.

[9] "VRML plugin and browser detector", http://cic.nist.gov/vrml/vbdetect. html/

[10] "Cortona VRML client", http://www.parallelgraphics.com/products/cortona/



치-시안 양 (Chi-Shian Yang)

- 2001.5~2005.5 말레이시아 멀티미디어 대학교 멀티미디어시스템전공(Bachelor IT)
- 2005.6~2006.2 멀티미디어대학교 시스템 분석엔지니어
- 2006.3~현재 동서대학교 디자인&IT전문대학원 유비쿼터스 IT학과 석사과정
- 주관심분야 : 유비쿼터스 컴퓨팅을 위한 사용자 인터페이스 환경, 컴퓨터 그래픽스

정완영 (Wan-Young Chung)

- 1992.8 경북대 전자공학과(공학석사)
- 1998.5 큐슈대 총합이공학연구과(공학박사)
- 1999.3~현재 동서대학교 컴퓨터정보공학부 부교수, 디자인&IT 전문대학원 원장
- 2004.7~현재 유비쿼터스 IT 전문인력양성 사업단장
- 2006.3~현재 BK21 WSN을 활용한 u-헬스케어 기술개발핵심사업팀 팀장
- 주관심분야 : 유비쿼터스 헬스케어, 무선센서네트워크, 반도체센서, 홈네트워킹