Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/081L24

The Implement of Interior 3D Navigation based on VRML

Jintong Rao[1,*], *Jianbing Xiahou*[2,*], *Yuan Li*[1] *and Yang Mu*[2]

[1] School of Architecture and Civil Engineering, Xiamen University, Fujian, China
[2] Software School, Xiamen University, Fujian, China

Received: 30 Apr. 2013, Revised: 25 Aug. 2013, Accepted: 26 Aug. 2013
Published online: 1 Apr. 2014

Abstract: With the rapid development of computer software and hardware, the virtual reality is applied more widely. Interior 3D navigation, being as an important part of virtual reality and especially for 3D virtual roaming system, has become a hot research topic in recent years. In this paper, we propose a novel method to implement interior 3D navigation based on Virtual Reality Model Language (VRML). Firstly, the 3D tree routing map of an office building is obtained from the accurate AutoCAD model and with the support of block management strategy. Then, the roaming path by waypoint navigation method is generated. Finally, the interior 3D navigation system based on VRML is developed. By the automatic navigation, user can realistically experience the 3D indoor scene and the results test the validity of the proposed method in the engineering project.

Keywords: 3D navigation, interior roaming, surface modeling, VRML

1 Introduction

In recent years VR (Virtual Reality) is a hot topic within domestic and foreign scientific and technological research areas, and its development is also changing quickly. VR technology has the I^3 (*Interactivity, Immersion and Imagination*) characteristics. Among them VRML (Virtual Reality Model Language) is the representative of Web3D, and it is used to construct an interactive virtual reality for multiplayer.

Virtual roaming is a virtual reality systems indispensable function, and especially Interior 3D navigation is an important constituent of the 3D virtual roaming system. Virtual roaming has been studied by many domestic and foreign universities and become a hotspot research area.[1,2,3,4,5,6,7,8,9,10,11]

A novel method about interior roaming system is presented in this paper. Through using the Surface Modeling function of AutoCAD (Autodesk Computer Aided Design) and prototype definition of VRML, we reduce the number of triangle surface in model effectively; While describing the interior traffic routes precisely by AutoCAD and combining VRML programming scripts, we can realize self-help interior 3D navigation capabilities.

This paper consists of five parts. The first one is introduction, we introduce a solution of interior navigation based on VRML through integrating the recent research home and abroad three dimensional interior navigation; In the second part we discussed how 3Dmax generate VRML file based on group management, combining Block management and 2.5D surface modeling of AutoCAD; In the third part, combining with idea of the VRML prototype, we give the corresponding prototype definition for common space unit in building, such as room, to improve network transmission efficiency; In the fourth part we describe the path of interior transport with AutoCAD accurately, give data structure definitions for the path tree, also define the prototype of virtual roaming navigation system based on VRML; In the last part, through the communication of VRML and HTML (Hypertext Markup Language), we define and develop Web3D-based interior 3D navigation system.

2 Block management and 2.5D surface modeling of AutoCAD

Experiments show that eyes will have maladjustment and fatigue obviously if the refresh rate of virtual environment is less than 20 frames per second and virtual reality will not conform to the factor. The more polygonal faces, more verisimilar the model is, and the heavier the system

* Corresponding author e-mail: raojintong@xmu.edu.cn, jbxiahou@xmu.edu.cn

burden is. So ensuring the verisimilitude of the scene, we have to reduce the number of faces possibly to meet the real-time requirement of the system. Currently most VR application system is established on the basis of the 3D model by 3Dmax software. But in this paper we use the Surface Modeling function of AutoCAD and import the model into 3Dmax so that the number of faces efficiently can be reduced.

As the core product of Autodesk, the world leading design software company, AutoCAD has become one of the most popular computer aided design softwares among engineers and technicians. The most important features of AutoCAD software are open system architecture, and high data accuracy, and it has two categories in Surface Modeling, 2.5D Surface Modeling and 3D Solid Modeling. There will be many redundant faces in 3D Solid Modeling which have to be optimized in virtual reality applications. While 2.5D Surface Modeling only builds the faces we can see, and because of its accuracy, it is conducive to programming applications in the VR system. On the other hand there is a graphical data management model in AutoCAD, Block, which is the number of combined elements (Drawing Line, Text, Label, etc.). After being imported into 3Dmax the Block will be converted to a group, and they generate the appropriate grouping transformation (*Transform Node*) by import groups into VRML. It is conducive to further scripting.

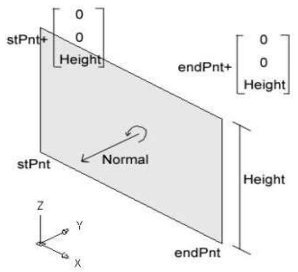

Fig. 1: Tessellate of the line

With the Web3D VRML-based engineering practice, we use AutoCAD's 2.5D Surface Modeling to build three-dimensional architectural model, finally minimize the number of triangles in three-dimensional model and improve the efficiency of virtual reality scenes and quality.

The 2.5D modeling refers to a single line to indicate the direction by thickness of the surface, not as same as Solid Modeling. The straight line in the AutoCAD has thickness of the property, representing the height of line. The height of line will be converted into the mesh model when imported into 3Dmax, and it has a normal direction. As shown in Figure 1, the normal direction complies with counter-clockwise rule (Right-Hand-Rule). The *stPnt* is the starting point while drawing the line, the *endPnt* is the end point, and the Height indicates the line thickness.

In interior virtual reality system, there are a large number of a size rooms, and we make a block to model these. Figure 2 shows different levels of an office building in the most common breakdown. From left to right there are windows, room unit, standard floor, and whole building. All faces in the model comply with the right hand rule. The faces toward the surface are visible normally. All the faces are single-sided, and make the block. In figure 2, the left-most diagram represents an enlarged window of the room, and the block named *Win900* indicates the windows width is 900mm. The insertion point is the lower left corner vertex. While the next one is a diagram of a floor room in the standard cell, and the block called *Room3500*, shows the width of the room is 3500mm. The insertion point is the left bottom corner vertex too. The third one from left is one layer of the building, and the block is named *Floor*. While the rightmost diagram is a three-dimensional model for the whole building, and the block is named *Office*.

As mentioned earlier, three-dimensional model in AutoCAD uses a hierarchical block management (shown in the right diagram of Figure 3). In this model one block can be included in another one, layers are nested, and a large model includes small models. After these models are imported into 3Dmax, they will be converted to a corresponding set of managements (shown in the right diagram of Figure 3). It has the same hierarchical group management, group in group, nested layers, and a large group containing small groups. Similarly, when exported to VRML file, three-dimensional models are converted to the corresponding hierarchical group management (shown in the left diagram of Figure 3), group containing group, nested layers, a large group containing small group. The conversion process is shown in Figure 3, in which AutoCAD block name, 3Dmax group name, VRML group are one-to-one corresponding.

3 Application of VRML prototype

Blocks application in AutoCAD is to combine repeated graphical portions into one graphic set so that they will be managed and modified easily. Generating groups in 3Dmax also aims to facilitate the modeling, and in VRML we can convert the block (group) to the prototype by programming to improve the running efficiency.

VRML language defines users own object through the use of prototypes (PROTO/EXTERPROTO). The prototypes have object-oriented class properties, packaging corresponding data structure and method, instantiated in the application, and certain scalability. Through the routing and script language programming, we can implement a WYSIWYG (What You See Is What You Get) visual user interface to build the scene, both for the convenience of the user operations and improving efficiency [12,13,14,15].

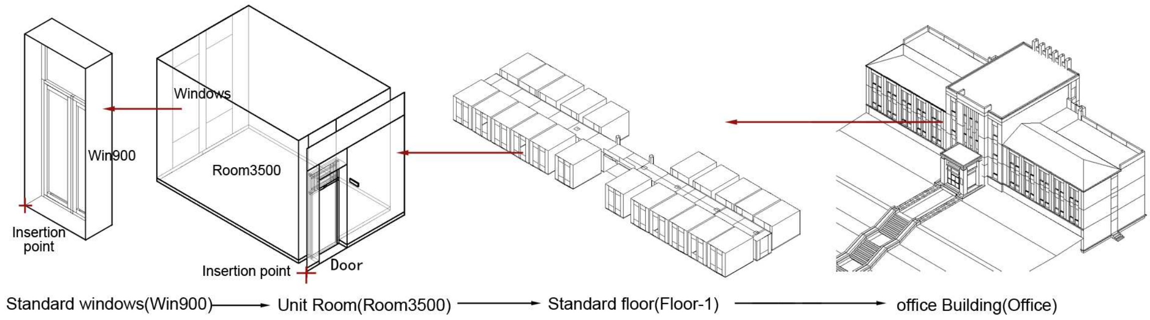

Fig. 2: The structure chart of office building model

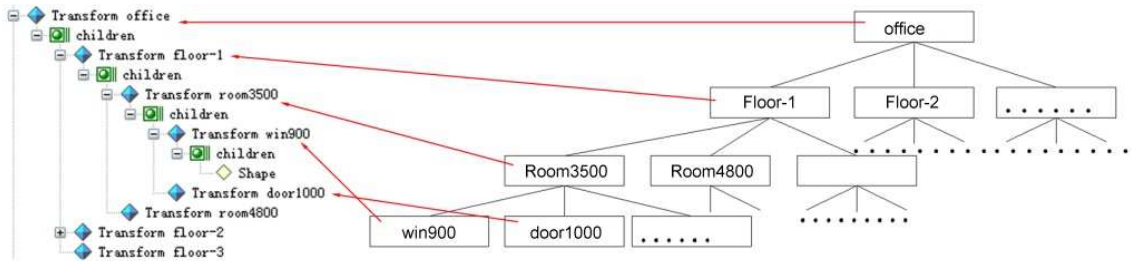

Fig. 3: The translation chart of hierarchical structure of 3D model

The most important element is the unit room. The requirement analysis shows that, as a prototype of the unit room (or class), we must analyze its data structure and included methods, which gives its prototype definition.

(1) First we need an insertion point (SFVec3f *position*), corresponding to a rotation angle (SFRotation *rotation*) and a scale (SFVec3f *scale*);

(2) Second, we need the room number (MFString *roomNo*), corresponding to the room name (MFString *roomName*);

(3) On-off status of the room door (SFBool *doorOpen*). Of course, there is a function to open doors, also we can open it by clicking on the door, or set a proximity sensor to open doors automatically.

(4) On-off status of the room light (SFBool *lightOnOff*); similarly there is a function to turn-on the light. In order to save resources, we set a proximity sensor. Light is on when someone walks into the room, light is off when he or she leaves the room.

(5) A network link will be set up (MFString *httpUrl*). A new network link as the introduction of this room can be the user's personal homepage, room's feature description web pages or other functions.

In conclusion, we give the definition of VRML prototype as follow:

```
PROTO room3500 [
    field SFVec3f translation 0, 0, 0        # position
    field SFRotation rotation 0, 1, 0 α
        # rotation angle α
    field SFVec3f scale 1, 1, 1        # scaling
    field MFString roomNo “”
        # the Number of the room
    field MFString roomName “”
        # the name of the room
    field SFBool doorOpen false # door on-off status
    field SFBool lightOnOff false # light on-off status
    field MFString httpUrl []
        # corresponding web links
]
{#main program body }
```

When the prototype is given, rooms in the building are instantiated in the application. Of course, we can get the positional parameters through the corresponding block's property values (the insertion point, rotation value, and scale ratio) in CAD, or appropriate corresponding Transform node in the VRML code. It not only reduces the amount of code, but also can be modified easily.

Similarly for other unit rooms, windows and other components, firstly we generate the prototype definition, then instantiate it, and at last generate the resulting WRL code file for the whole office building. It is obvious that the size of code is reduced and we can greatly accelerate the transmission speed and improve online operating efficiency [16].

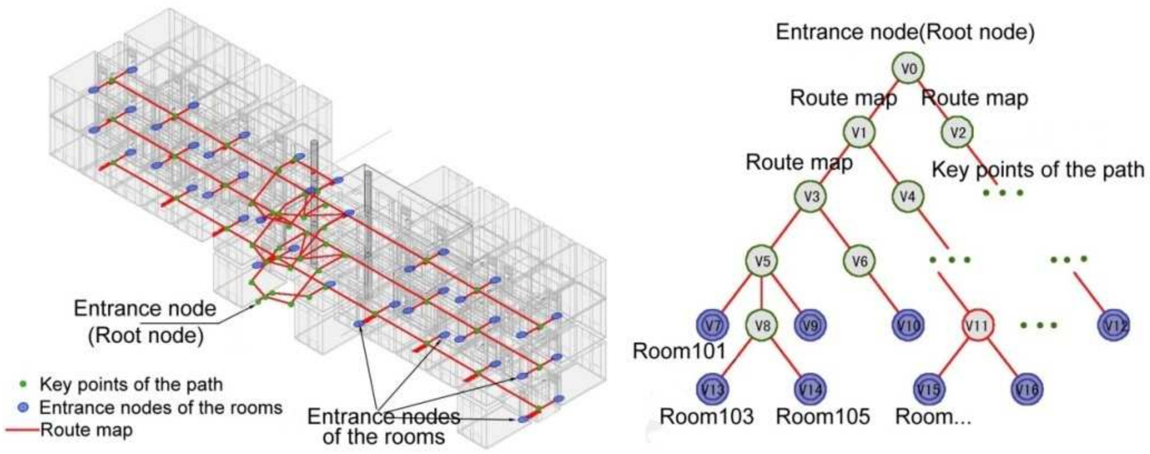

Fig. 4: The map and tree structure of interior route

4 Implementation of three-dimensional navigation system

In order to generate the self-help navigation roaming system for the virtual reality of the office building, we must obtain the office buildings three-dimensional routing map firstly, generate roaming path by waypoint navigation method, and then implement the three-dimensional navigation.

4.1 Interior three-dimensional route map

The original three-dimensional model of the system is created in AutoCAD. Considering the accuracy of AutoCAD, we can get the three-dimensional route map of the office building in AutoCAD.

When visiting the office building, users start from the entrance floor. The most important is how to determine a route from the floor at the entrance to the office to be visited.

Firstly we draw the whole building traffic route map, assuming that we are in the central axis of the traffic area location, such as aisles, stairs; Secondly we paint the entrance traffic line at each room entrance and let them intersect with central axis of the corridor. The detail is shown in Figure 4.

Left diagram in Figure 4 illustrates the three-dimensional interior traffic routes. We define the entrance node at the entry. And the node will also serve as the root of its tree structure, in which small dots (green) are the key points of the path, corresponding to the green box of the right tree node; while the large dots (blue) in the left diagram are the entrance nodes of the rooms, corresponding to the tree leaf node (blue) in the right diagram. For example V_7 indicates *Room101*, and V_{13} indicates *Room103*.

4.2 The data structure of the route map tree

We find it is clear that each leaf node indicates a destination, a room or a special place indoor. Therefore, the navigation system will transform to the Preorder tree traversal, which finds its destination and the traversal path of the leaf node. For instance if one user would like to go to *Room105* from the entrance, we finally determine the route to be $V_0 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_8 \rightarrow V_{14}$ by tree traversal. In order to get the route data of the system, we define the data structure of tree node in Figure 5. The data field contains two data: three-dimensional coordinate position (x, y, z) and spatial point's ID. For example, V_{12}'s ID value is “105”, which corresponds to *room 105*. Pointer field points to the next node, in other words, the next spatial point.

By tree traversal we can get the route point from the entrance to a specific location, the point series of three-dimensional coordinates denoted as MFVec3f tourPoint[]. Through these points, we can generate roaming routes easily.

4.3 Navigation system

The essence of the virtual roaming is to set a series of viewpoints (also known as the camera) to the key position (frame), and to calculate the time for each route. The purpose is to walk in uniform speed. Then we set the time interval by time sensors, generate the two camera positions of two key frames with the corresponding interpolator, generate a series of animation and finally achieve the user's roaming process.[17]

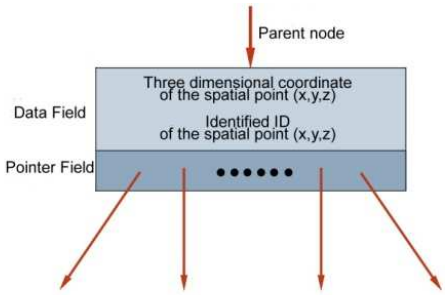

Fig. 5: The structure of tree node

Through analyzing the grammatical structure of viewpoint, there are three basic attributes: position, orientation and focal length (fieldOfView). The direction is a three-dimensional direction vector, and in order to facilitate the description and implementation, it can be decomposed into two angle(α,β), assuming that the focal length is same, the generation of view point can be decomposed into the following four steps:

Firstly the default location of viewpoint is the origin (0, 0, 0) and the default direction is on the negative **Z** axis (0, 0, -1);

Secondly, to move the viewing directions around (1, 0, 0) on the **OX** axis and α degrees, equivalent to the people's heads rise up and down;

Thirdly to move viewing directions around the **OY** axis and β degrees, equivalent of turning around the human body;

Finally, to move the camera from the original position to the final position.

Transform node of VRML language has three fields: Transform, Rotation, Scale, as the multiply between the transform matrix does not have the commutativeness, while **TRS** rule is obeyed in three-dimensional transform in VRML, so scale (**S**) firstly, then rotation (**R**) ,then translation (**T**). Since the camera has been moved by two rotation shift and one transform, as the program structure of the roaming camera node shown presented soon afterwards, it can achieve the conversion process above.

Therefore, above all we must define the architecture of a roaming viewpoint node, and through the analysis of viewpoint structure in VRML, we define the framework of viewpoint node as follow:

```
DEF autoView Transform {
    translation x, y, z        #translate to(x, y, z)
    rotation 0, 1, 0,β         #rotate β around OY
    DEF autoViewPoint ViewPoint {
        orientation 1, 0, 0,α
            # rotate α around OX
        fieldOfView 0.785
            #focal length of viewpoint
        description "autoNavigation"
    }
}
```

It consists of three parameters: viewpoint position (x, y, z), the direction of camera (α,β), and the initial direction of the camera is (0, 0, -1), negative to Z axis. To determine the direction there are two steps: rotate α around **OX** first and then β around **OY**.

According to engineering experiment, we set the following program framework:

```
DEF vpTimeSensor TimeSensor {cycleInterval t}
#time period sensors, period is t
DEF viewP PositionInterpolator {Key [] keyValue []}
ROUTE viewPos.value_changed TO autoView.translation
#change the position of viewpoint
DEF viewD OrientationInterpolator {Key [] keyValue []}
ROUTE viewD.value_changed TO autoView.rotation
#change the angle of viewpoint in plane XOZ
DEF viewUD OrientationInterpolator {Key [] keyValue []}
ROUTE viewUD.value_changed TO autoView.rotation
#change the viewpoint in the horizontal direction
```

We set three interpolators for ($position,\alpha,\beta$) and share the same value of *key[]* to indicate same time intervals. To ensure data consistency, each key is set to point to the direction of the next point. Therefore the keyValue of the viewP is the *tourPoint []*, which is a series of three-dimensional coordinates. While the keyValue of the rotation interpolator viewD is a series of $[0,1,0,\beta]$, and the keyValue of the rotation interpolator viewUD is a series of $[1,0,0,\alpha]$.

4.4 Data generation of tour system

The keyValue of the three interpolators represents the camera attribute values of key points($position,\alpha,\beta$), indicating the camera position. The direction of first turning around **OX** axis for α, and then around **OY** axis for β, which all come from the output vertex set of polyline MFVec3f *TourPoint []*. And it contains polyline data of converted arc, besides the position of each vertex are as a key point during the tour processes.

Step1: To calculate the time list value of the key points (*key []*). Firstly, to find out the total length of the polyline, assuming that people's walking speed $v = 1m/s(3.6km/h)$, we can obtain the total time t which is cycle time of tour, set t as vpTimeSensor.cycleInterval value of the cycle time sensor, then we can give the time list value in the *keyValue []* accordingly .

Step2: We pass *TourPoint []* value to the viewP.keyValue, which is the position value of key points.

Step3: To determine the direction of the camera; we assume the default direction of each key point is go to the

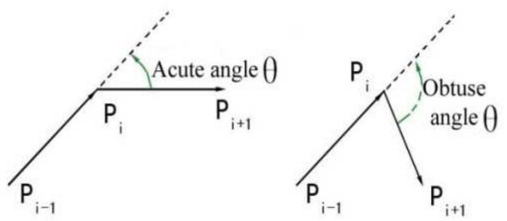

Fig. 6: The rotate angle of a line segment

next key point (equivalent to the tangent direction), the vector direction of the key point is $(P_{i+1}-P_i)/\ P_{i+1}-P_i$, then converted into (α,β) , we need which is discussed previous by related the matrix operations. According to this definition, the direction is changing in addition to position constantly in time while touring between the two key points. Considering the situation shown in Figure 6, when the user walks from the point P_{i-1} to P_i, the rotation angle θ which is less than the threshold θ_0 in the left diagram of Figure 6(angle threshold can be assumed as $\theta_0 = 90$ degrees), is an acute angle, the direction of change in motion is gentle relatively. But as shown in the right diagram of Figure 6, the rotation angle θ which is greater than the threshold θ_0, is obtuse angle, the direction changes faster, and sometimes it produces a jump effect. So we can allow users to walk from the point P_{i-1} to P_i without changing the direction , then settle in the P_i point to do not move with rotating angle θ slowly in the same position.Correspondingly we increase one key point (the two P_i points); time series value should be adjusted accordingly at the same time.

After finishing calculation of the roaming path data, we can generate the VRML codes by combining the definition of autoView and the three interpolations, which ultimately generate the implementation and visualization of the tour system.

5 Architecture of roaming system based on Web3D

VRML, a three-dimensional modeling language used in web browser, is the technology standards of Web3D. Though VRML does not have the communication function itself, we can program that through its script to support network communication. As a result the VRML node can receive data through the network and can render the scene in real-time, so we can create a communication between HTML and VRML to build the roaming system based on Web3D shown in Figure 7. Considering for the limitations of VRML language, interface is designed by the web. By stimulating the menu on the page, we pass the events to roaming nodes in the VRML system to traverse the tree path. And we can create path data of real-time roaming, result in roaming effect.

There are two aspects of data streams in the communication between HTML and VRML: Command Stream and Parameter Stream. VRML module sets up a control center (eventControl) to manage all eventIn and eventOut. When receiving the directives issued by the HTML, the eventIn will be triggered. And then it produce an eventOut to control part of the module through the script processing; when modeling module completes the tasks ,it will send resulting data related through Browse.LoadURL directively by transmitting OnEvent to VRML module:

Browser.LoadURL (“OnEvent (*Type*, *Info1*, *Info2*)”, “”) where *Type* indicates the type of information, and *Info1*, *Info2*, indicate the properties of objects.

HTML module establishes an Information Receiving Center: <Script for=“*vrAcitve*” event= “OnEvent (*Type*, *infoList*)” language=“Javascript1.2”>, corresponds to the sending commands of the VRML information, and processes according to the received information; While receiving and sending the relevant data initiatively by setEventIn and getEventOut method:

Document.getElementById (“*vrAcitve*”).setEventIn (*nodeName*, *fieldName*, *value*);

Document.getElementById (“*vrAcitve*”).getEventOut (*nodeName*, *fieldname*);

Fore mentioned *vrAcitve* is the ID of ActiveX rendering plug-in in HTML, and *nodeName*, *fieldName* represent nodes name and domain name in VRML.

Obviously the mechanism, through which VRML controls data flow, is to send initiatively and to receive something passively. But the HTML is different; its data flow control mechanism is to receive initiatively and to send something initiatively. For this reason, storage and control of path tree can be developed only in HTML part. So we can ensure the safety and efficacy of system data.

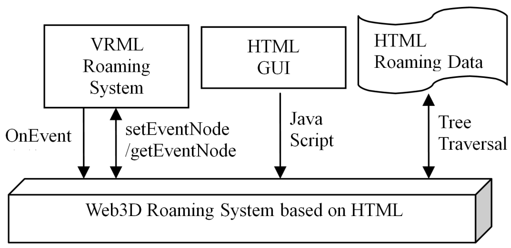

Fig. 7: The Flow chart of Roaming system

6 Conclusion and Outlook

In this paper we take the common interior three-dimensional navigation system for example. With modeling in AutoCAD, we provide an accurate interior traffic route diagram, generate tree route map, combine with the corresponding prototype definition of VRML, and develop a three-dimensional intelligent navigation system quickly. The method has been applied in a number of projects, and has been applied with great practical significance, and has improves engineering efficiency greatly. This method also applies to outdoor three-dimensional intelligent navigation system, in which the traffic route diagram is complexer, and the route map will be the shape of topology “map”. The algorithm also has to be modified slightly, which is what well continue to research on.

Acknowledgement

This work is supported by National Natural Science Foundation (40801158), Fujian Natural Science Foundation (2010J05099), and Universities Creativity Foundation of Xiamen (3502Z20093003). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Y. J. Xi, X. L. Li, Y. Duan and N. Maerz, Virtual Navigation of Interior Structures by Lidar. In Proc. SPIE, Orlando, Florida, USA, (2011).
[2] B. Nicholas, D. Pratik and R. Kuldip, Development of a tightly coupled wireless acoustic sensor network aided inertial navigation system for indoor navigation, In 23rd Int. Tech. Meet. Satell. Div. Inst. Navig., Portland, OR, USA, **4**, (2010).
[3] D. Cheng, Y. Peng, Study in interactive virtual campus based on VRML, In Int. Conf. Comput. Appl. Syst. Model., Taiyuan, Shanxi, China, **4**, (2010).
[4] W. M. Tsai, L. T. Hsu and S. S. Jan, The development of an indoor location based service test bed, In 22nd Int. Tech. Meet. Satell. Div. Inst. Navig., Savannah, GA, USA, **6**, (2009).
[5] H. R. Chen, R. C. Peng, S. J. Li and C. X. Yu, The Visualization of 3D Terrain Based on VRML, Information Technology and Applications, **3**, 116-118 (2009) .
[6] I. Lucio, R. Roberto and C. Luca, Automatic derivation of electronic maps from X3D/VRML worlds, In Web3d Symp. Proc., Monterey, CA, USA, (2004).
[7] W. Y. Chung and C. S. Yang, Dynamic VRML-based navigable 3D map for indoor location-aware systems, In Lecture Notes in Electrical Engineering, **21**, 269-284 (2008).
[8] S. Liu, Research and implementation of virtual reality scene roaming technology based on VRML, Msc Thesis, Ji Lin University, (2007).
[9] T. S. Mu, Research and implementation of roaming technology based on virtual reality, Msc Thesis, Wuhan University of Technology, (2005).
[10] J. C. Wang, J. Q. Luo and S. S. Zhang, Research on Getting 2D Navigational Maps of the VR Scene, Journal of System Simulation, **16.7**, 1499-1501 (2004).
[11] S. Vitor, N. Kia, W. Erik, Goncalves Joao G.M. and H. David, Automated 3D reconstruction of interiors with multiple scan-views, In Proc .SPIE, San Jose, CA, USA, **3641**, (1998).
[12] X. Y. Duan and P. Jiang, Exploration of Virtual Avatar Technology Based on VRML and Java, In Int. Workshop Knowl. Discov. Data Min., Moscow, Russia, (2009).
[13] C. S. Yang and W. Y. Chung, 3D indoor activity monitoring navigation viewer, In Int. Conf. Convergence Inf. Technol., Gyongju, Korea, (2007).
[14] G. Z. Feng, J. B. Chi, C. Wang and D. M. Wang, Design and Implementation of Visual Interactive Manipulation in Model Builder of VRML, Journal of System Simulation, **18.2**, 387-390 (2006).
[15] K. Szilrd and N. Anton, Viewpoint adaptation during navigation based on stimuli from the virtual environment, In Web3d Symp. Proc., Tempe, AZ, United states, (2002).
[16] S. Mohsen, G. F. Hashemi and E. Mehdi, A new fractal-based approach for 3D visualization of mountains in VRML standard, In Proc. GRAPHITE Int. Conf. Comput. Graph. Interact. Tech. Australas. S. East Asia, Singapore, Singapore, (2004).
[17] J. B. Xiahou, J. T. Rao, Y. Li and Y. Yao, Visual Implement of Virtual Reality Tour based on VRML, Applied Mechanics and Materials, **48-49**, 1351-1357 (2011).

Jintong Rao received MS degree in computer science from Software School of Xiamen University. He is a Senior Engineer at the School of Architecture and Civil Engineering(SACE), Xiamen University, also dean of Architectural Digital Design Lab. His principal fields of interests are 3D parametric modeling, BIM applications and Virtual Reality Applications.

Yuan Li is an associate Professor at the School of Architecture and Civil Engineering (SACE), Xiamen University. He graduated from Wuhan University and got his PhD degree in photogrammetry and remote sensing. His principal fields of interests are Urban Planning Support System, 3D GIS, GIS, Routing Navigation, and Emergency Response.

JianBing Xiahou received bachelor Degree in Electronic engineering, Master of Science Degree in Radio physics and PhD in finance from Xiamen University. He is currently an associate professor of Software School, Xiamen University.The provincial and municipal Science Foundation, research funds which he presided over are more than one million in total. His research interests are in the areas of Virtual Reality, Embedded System and Financial Informatization.

Yang Mu received the MS degree in computer from Xiamen University. He is currently a Student in Department of Software at Xiamen University. His research interests are in the areas of 3D graphics and Virtual Reality.