2차원 Map 기반 3차원 가상공산 Navigation
(2D Map-Based Navigation in 3D Virtual Environment)

권태욱[†] 최윤철[††]
(Tae-Wook Kwon) (Yoon-Chul Choy)

요 약 이 논문에서 제시된 2D Map-Based Navigation (MBN)은 다중 사용자 환경의 가상공간을 이동할 때, 실세계와 유사한 이동방법을 제공하여 사용자로 하여금 가상공간에 대한 현장감 및 현실감의 제고에 초점을 두었다. MBN은 사용자들이 가상공간에서 발생하기 쉬운 spatial loss를 방지하고, 이동 시 부가적인 입력이 없이도 일정한 속도로 이동을 지원하는 Automatic Constant-velocity Navigation, 이동 중 장애물 및 다른 사용자(아바타)와의 충돌현상을 감지 및 회피하는 Collision Detection and Avoidance, 그리고 충돌회피 후 기존 방향으로의 계속된 이동을 지원하는 Path Adjustment 등의 기능을 제공한다. MBN은 spatial loss의 방지, 사용자의 부가적인 노력의 감소 및 병행작업의 보장, 현실과 유사한 사용자 중심의 navigation 기법의 제공, 그리고 가상공간과 현실과의 괴리를 줄임으로써 가상현실이 추구하는 현실감 및 현장감을 높일 수 있도록 하였다. 실험을 통하여 본 연구에서 제안한 MBN이 사용자 중심의 매우 자연스럽고, 쉽고 편리한 가상공간 navigation 인터페이스라는 평가를 얻었다.

Abstract This paper focuses on the navigation in the virtual environment, one of the major interfaces for the interactivity between human and virtual environments in virtual reality circumstances and worlds. It proposes a new navigation method: 2D Map-Based Navigation (MBN), which can prevent user's spatial loss in 3D Virtual Environments and provides a very simple, natural and user-centered navigation method in virtual environment which can improve the sense of presence and Reality. The MBN is composed of three major processes: Automatic Constant velocity Navigation, Collision Detection and Avoidance, and Path Adjustment. The MBN can reduce user's difficulties and improve user's sense of presence and reality in the virtual environments. Through the experimental study it has been proven that the 2D Map-Based Navigation is a very natural, straightforward, and useful navigation interface in the virtual environment.

1. Introduction

An important feature of virtual reality is Navigation, the facility for the user to move through a virtual environment in a natural and easily controlled manner. It involves changing the perspective of the user in the Virtual Environment. It allows the users to move in the Virtual Environment as well as reorient themselves to look at the world differently. Natural locomotion methods can contribute to a sense of presence, which has been cited by some researchers as a defining attribute of VR. The illusion of presence can be lost through unnatural experiences during travel in VE. This can be caused by poor interaction metaphors or by experiences, which do not agree with the user's everyday understanding of the real world. Several attempts have been made to develop new metaphors for walking through virtual environments. However, the intuitive metaphors described so far only can solve some of the problems. The other part concerns how to provide a virtual environment with more realistic properties so that the user's movement can be more natural and comfortable.

No matter how powerful the technology of the

[†] 정회원 : 육군사관학교 전산학과 교수
kwontw@kma.ac.kr
[††] 종신회원 : 연세대학교 컴퓨터과학과 교수
ycchoy@rainbow.yonsei.ac.kr
논문접수 : 2000년 6월 23일
심사완료 : 2001년 2월 5일

virtual environment is, it is the user interface that determines how the system will be used. If people can only see and feel the virtual world without any interaction, then the application of this system would be very limited. Moreover, the interaction between the users and the virtual world can further enhance the reality of the virtual world. Thus, in addition to the creation of 3D virtual scenes, the interface procedure will also be designed in this project, so that users can navigate in this virtual world through basic input devices such as a mouse and keyboard. An intuitive interface between the user and the computer is one which requires little training and proffers a working style most like that used during the interaction has with environments and objects in the user's' day-to- day life.

This paper focuses on the navigation method in the virtual environment, which is one of the major interfaces for the interactivity between human and virtual environments in VR circumstances and worlds. It proposes a new navigation method: 2D Map-Based Navigation, which prevents the user's spatial loss in 3D VE and provides a simple and natural navigation method in virtual environments, closer to how we walk or move in the real world. It can then improve the Presence and Reality, which is the final goal for Virtual Reality. The MBN periodically displays the user's current position on the map as a black dot for the user's spatial awareness. The MBN is composed of three major processes. If the user selects a point on the 2D map then the relative target position on the 3D VE is estimated. Then the Automatic Constant- velocity Navigation (ACN) process supports continuous and automatic constant travel and navigation service until the user arrives at the selected target location, without any additional input from the user. The second process is the Collision Detection and Avoidance (CDA), and it consists of two steps: Collision Detection and Collision Avoidance. The first step detects and recognizes a collision situation with virtual objects or other avatars in the virtual environment. The second step avoids a collision situation with a virtual object or any other user's' avatar, by systematically supporting the user's avatar by-passing the object or the other user's avatar without stalling and passing through. The last process is Path Adjustment, which supports the user's avatar to maintain it and return it to its original travel and navigation direction after the Collision Avoidance step.

2. Related Works

The most common task in VE's is that of navigating around the space of the environment. Some artificial methods must be provided for the user to move through the space, such as walking, artificial flying, and physical user motion with treadmills, roller skates, bicycles, etc [1]. A number of researchers have addressed the issues related to navigation and travel in both the immersive virtual environments and in general 3D-computer interactions. They have insisted that studying and understanding human navigation and motion control is of great importance for comprehending how to build an effective virtual environment travel interface.

Bowman [2, 3] presents a categorization of techniques for the first-person motion control, or travel, through immersive virtual environments, as well as a framework for evaluating the quality of different techniques for specific virtual environment tasks. Results indicate that "pointing" techniques are relatively more advantageous compared to "gaze-directed" steering techniques for a relative motion task, and to those motion techniques which instantly teleport users to new locations and are correlated with increased user disorientation.

Various metaphors for viewpoint motion and control in 3D environments have also been proposed. The flying, eyeball-in-hand, and scene-in-hand metaphors for virtual camera control are identified [4, 5, 6]. As an extension of the scene-in-hand metaphor, Pausch et al. [7] makes use of a World-in-Miniature (WIM) representation as a device for navigation and locomotion in immersive virtual environments. Mine [8] offers an overview of motion specification interaction techniques. The overview also discusses the issues concerning the implementation of such specification in *immersive virtual environments*. Several user studies concerning immersive travel techniques have been reported in the literature, such as those comparing different travel modes and metaphors for specific virtual environment applications. Physical motion techniques have also been studied, including an evaluation of the effect of a physical walking technique on the sense of presence [9, 10].

"Wayfinding" issues have been the subject of studies by Darken and Sibert [11, 12]. The use of maps, breadcrumbs, and landmarks were evaluated as tools for finding a path through a virtual environment. Their research shows that subjects in the treatment without any additional cues were often disoriented and had extremely difficult to complete the task. For effective navigation, the results suggest that users of large-scale virtual worlds require structure, and augmentations such as direction indicators, maps, and path restriction which can greatly improve both the wayfinding performance and the overall user satisfaction.

Satalich [13] studied the navigation and wayfinding in virtual reality environments. The tasks assigned to the subjects for investigation of navigational awareness utilized the following three measures: orientation, route estimation and Euclidean estimations. The results indicated that having a map before entering the virtual environment can improved the performance, but not for exploring the virtual environment.

In addition, Xiao [14] presents a new technique for controlling a user's navigation in a virtual environment. It introduces artificial force fields, which act upon the user's virtual body such that the user is guided around obstacles, rather than penetrating or colliding with them. Li [15] describes an auto-navigation system, in which several efficient path-planning algorithms adapted from robotics are used.

3. 2D Map-Based Navigation (MBN)

An important feature of virtual reality is Navigation, the facility for the user to move through a virtual environment in a natural and easily *controlled manner*. Natural navigation methods can contribute to a sense of presence, which is cited by some researchers as a defining attribute of VR. The illusion of presence can be lost through unnatural experiences during travel in VE. This can be caused by poor interaction metaphors or by experiences, which do not agree with the user's everyday understanding of the real world.

When the user travels in 3D VE, sometimes the user cannot easily figure out their location in the 3D virtual environment. We name this "spatial loss". Spatial loss is one of the major problems in the virtual reality system. In order to reduce spatial loss, the Virtual Reality service developers consider the user's spatial awareness. Spatial Awareness is how well we perform in the virtual world, or under experimental conditions using our spatial ability. Spatial Ability is composed of various dimensions. The *three major dimensions of spatial ability that* are commonly addressed are spatial orientation, spatial visualization and spatial relations. Spatial ability is usually measured by psychometric tests. Environmental enhancements such as signs, grid-lines, and directional arrows represent one major category of artificial visual navigational aids. Such methods probably will enhance navigation performance in virtual worlds. It may be also able to enhance ones spatial awareness by advancing toward more human and ecologically proper integration of multimodal stimulation and natural interaction, which will enable us to travel in virtual worlds just as well as we can in the real-world.

In most of the existing desktop Virtual Reality systems, if the user wants continuous travel or navigation, then the user has to input each moving *event continuously with a mouse or keyboard*. To control the navigation in 3D VE by mouse, keyboard or other input devices is not an easy job for a typical VR user; especially when the user is not familiar with VR or the computer environment. Also, when we travel in 3D VE, spatial loss can happen because it is not easy to recognize our location in 3D Virtual Environments without any additional informations.

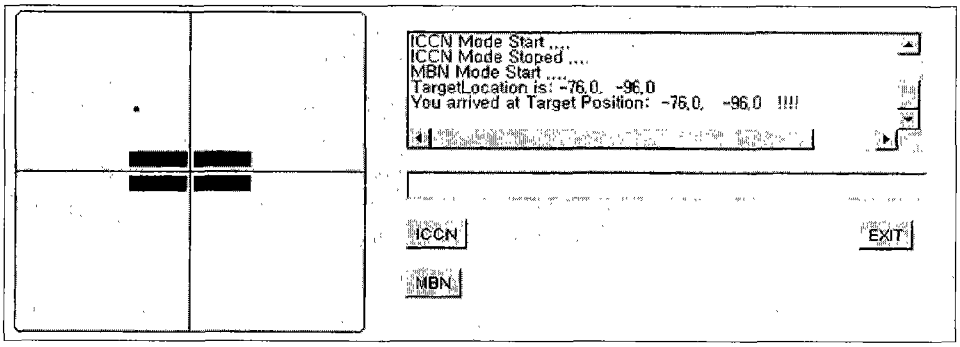
Furthermore, when the avatar runs against any object in the virtual environment, the system generates a collision detection event and then the avatar is stalled and is prevented from moving any further. And if the avatar meets with any other avatar in the multi-user virtual environment then the two avatars go through each other, because the existing VR systems do not support the collision detection between two avatars. These situations are excessively different from our real-living situations, and cause reduction of the Presence, Reality, and Immerse feelings for the VR users.

The 2D Map-Based Navigation (MBN) can solve such problems in the existing VR service systems and improve Reality and Presence for users, supporting navigation method similar to the real world and gives 2D-based location information for user's spatial awareness. In order to improve spatial awareness for user, in the MBN (see Fig. 1), the system periodically displays the user's current position on the map as a black dot. When the user chooses the "MBN" button, if the user selects a point on the map then the system calculates the relative destination on the 3D VR space. Then it displays that *"target location"* message on the text window, and the user moving toward the selected target location, with support from the three major processes, which are described next. If the user move to the selected location then the system also displays the message: *"arrived at the selected target position"*.

The MBN provides a natural navigation method in virtual environments, closer to how we walk or move in the real world. It can then improve the Presence and Reality, the final goal for Virtual Reality. The MBN provides three major processes. The first process is the Automatic Constant-velocity Navigation (ACN), which supports continuous and automatic constant traveling and navigation service without any additional input from the user. The second process is the Collision Detection and Avoidance (CDA), and it is made up of two steps: Collision Detection and Collision Avoidance. The first step detects and recognizes a collision situation with the virtual objects or other avatars in the virtual environment. The second step avoids a collision situation with a virtual object or any other user's' avatar, by systematically supporting the user's avatar to by-passing the object or the other user's' avatar without stalling and passing through. The last process is the Path Adjustment (PA), which supports the user's avatar to maintain it and return it to its original travel and navigation direction after the previous Collision Avoidance process.

3.1 Automatic Constant-velocity Navigation (ACN)

In most of the existing desktop VR service systems, the user controls the speed of navigation



```
ICCN Mode Start ....
ICCN Mode Stoped ....
MBN Mode Start ....
TargetLocation is: -76.0, -96.0
You arrived at Target Position: -76.0, -96.0 !!!!
```

Fig. 1 Applet Window for 2D Map-Based Navigation and travel with a mouse, keyboard or other devices such as a bike pedal, glove, or joy stick. But if the user selects the MBN service, then it provides a Automatic Constant velocity Navigation (ACN) service to the user until the user arrives at the target location. ACN is an idea that came from the cruise control technique of a vehicle, which provides a continuous and constant driving speed service to the driver until the driver steps on the acceleration or breaking pedal. This driving service is very comfortable and useful, because it can reduce a driver's fatigue and effort when driving through a very long, straight and spacious road, such as an express way.

ACN supports this automatic constant and continuous travel as if the user is continuously giving moving events with various input devices. If ACN is serviced, it gives hands-free navigation. Hence, the user can handle other tasks as well, such as chatting, reading, or talking on the telephone, while traveling and navigating in the virtual environment, without inputting the moves in the VE. ACN operates in toggle mode, so if the user wants to turn off the ACN service, then the user simply needs to select the "MBN" button again. ACN function can be used in various situations, such as when the user wants to travel or walk in a very large and wide VE area or when the user wants to take a walk in the VE alone or with other user's avatars. This function will also be very helpful to novice or handicapped users, for they may have some difficulties in operating a mouse or keyboard.

In order to support ACN, we change the user's viewpoint by modifying the position variable in the viewpoint node in the VRML world file. At first, we calculate the angle (azimuth) between the target location and the user's current location, using the JavaScript Math.atan function with the location values of the user(X_o, Z_o) and the target(X, Z), angle (θ) = atan ((Z - Zo) / (X - Xo)).

We classify the angle(θ: 360 ^{0}) into 4 sections, and then continuously add to or subtract from the user's current position value in the viewpoint node, depending on the angle. For instance, if the angle is - 0.39249(- 22.5 ^{0}) then it conveys the plus sin(22.5) value to the X-axis value, the minus cos(22.5) value to the Z-axis value; in the user's current position variable in the viewpoint node. If the angle is + 0.39249(+ 22.5 ^{0}) then it conveys the minus sin(22.5) value to the X-axis value; the minus cos(22.5) value to the Z-axis value; in the user's current position variable in the viewpoint node. If the angle is 1.96245(+112.5 ^{0}) then it conveys the minus sin(112.5) value to the X-axis value; the plus cos(112.5) value to the Z-axis value; in the user's current position variable in the viewpoint node. If the angle is 3.53241(+202.5 ^{0}) then it simultaneously conveys the plus sin(202.5) value to the X-axis and plus cos(202.5) value to the Z-axis value in the user's current position variable in the viewpoint node. While supporting from ACN service, the user can also change his navigation direction, by giving direction change event with the typical input devices, like users do in a traditional virtual reality service system.

3.2 Collision Detection and Avoidance (CDA)

Collision Detection and Avoidance (CDA) process first catches and recognizes a collision situation with virtual objects or with other user's' avatars in the virtual environment. If the user's avatar runs against a virtual object then it generates a collision detection event, therefore the CDA has to systematically catch or receive this collision detection event, using virtual object's location data. It has to generate a collision detection event when the two different avatars approach each other also. To detect collision situation, the CDA uses the virtual object's and avatar's data information in the system, such as their location, scale data, and so on. If two different avatars approach each other with a distance less than 1~2m apart then this process recognizes that these two avatars will make a collision, and generates a collision detection event for the next step. After the Collision Detection, this process starts the next step: Collision Avoidance. This step assumes that the user's avatar will by-pass objects and other avatars systematically and naturally until the avatar is moved to a distant location, which will prevent colliding with the object or the other user's' avatar (see Fig. 2 and 3).

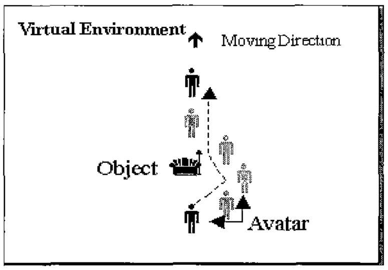


Fig. 2 CDA and PA with objects

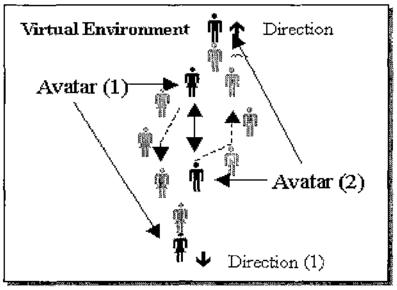


Fig. 3 CDA and PA with avatar

The direction of any bypass generally depends on how the user moves the user's avatar. However, the CDA is used for the avatar to move automatically to the right or to the left according to the object's relative location to the user's navigation direction. If the object is on the right side of the user's navigation direction, the bypass will move to the left, and vice versa. At this step, when the CDA receives or recognizes the collision detection event, it generates the right- or left- and forward- movement events making the avoidance paths look like an ellipse, until the collision detection event does not occur any more. The ellipse route is much more similar to a real life situation than any other shaped route, and the avatar's behavior is smoother. At this process the system counts the number of generated right/left and forward movement events for the next process.

The CDA process, the most time- and performance- consuming process in this research, uses the location data of other virtual avatars and objects in VE. It predefines the angle between the user and other avatar or the virtual objects based on the user's orientation (Direction of movement) (see Fig. 4). The angle between the user and an object A (θ_a) is calculated by the JavaScript Math.atan function with the location values of the user (X_o, Z_o) and the object A (X_a, Z_a).

$$\theta_a = \tan^{-1}\frac{X_a - X_o}{Z_a - Z_o},\ \theta_b = \tan^{-1}\frac{X_b - X_o}{Z_b - Z_o}.$$

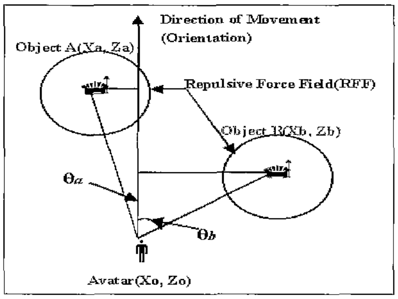


Fig. 4 Collision Detection

This collision detection process considers only the virtual objects, located in the focus area: less than 10 meters from the user's location. If a virtual object is in the focus area and the sin(θ) value is smaller than Repulsive Force Field (RFF)(1.5 times the object scale value), then the user's avatar and the object make a collision situation. In Fig. 4, the object A is in a collision situation because the sin(θ) value is smaller than the radius of the circle A (RFF-a). However, the object B is not in a collision situation, for the sin(θ) value is bigger than the RFF-b.

If the collision detection occurs, the process starts the next step, Collision Avoidance. The goal of this step is to make this process seems more natural and more similar to the real life situation when we come across an object or other people on the street. Moreover, we try to make this avoidance path look like an ellipse based on the scale value of the virtual object and avatar. The ellipse route is much more similar to a real life situation, so it makes the avatar's behavior smoother and more natural. The avoidance step starts from the position, 1.5 meters plus the object's RFF. And this avoidance path will bypass the position 1.0 meters away from the object.

The direction of avoidance is dependent on both the sin(θ) and the user's orientation value. When the user's orientation value is in between + 0.0 (+ 0 °) and + 4.70988 (+ 270 °) and the sin(θ) value is smaller than the user's orientation value, then we can assume that the object is on the right side of the user's locomotion way. Therefore, the left way collision avoidance is performed for this case. But when the user's orientation value is in between - 0.0 (- 0 °) and 1.56996 (- 90 °) and the sin(θ) value is smaller than the user's orientation value, we can assume that the object is on the left side of the user's locomotion way. Thus, the right way collision avoidance is performed in this case. At this step, it generates the right- or left- and forward- movements, which makes the avoidance path look like an ellipse. The system also counts the number of generations for right- or left- and forward- movements for the next process, Path Adjustment.

The process is the same as above, when the user's avatar meets with another avatar other than virtual objects, but is much simpler. When two avatar approach each other with a distance less than two meters from each other then it recognizes a collision situation and they simply bypass by moving one meter to the right or the left from the user's direction of locomotion depending on the other avatar's' location. The left way collision avoidance is performed, when the other user's avatar is on the right side of the path of user's locomotion, and vice versa. It generates right- or left- and forward- movements, which make the avoidance path look like an ellipse also (see Fig. 3).

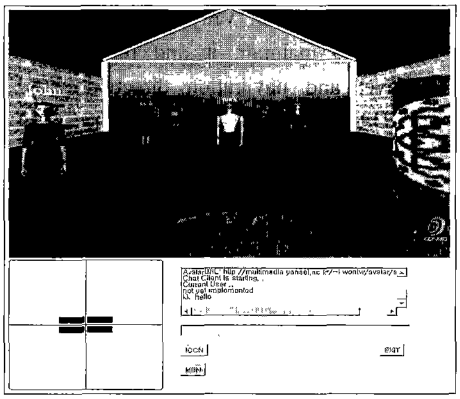
3.3 Path Adjustment

The goal of this final process is to guarantee that the avatar keeps his original travel/navigation direction and path, in order to reduce spatial loss while navigating in 3D VE. Which supports the user's avatar to maintain it and return it to its original travel and navigation direction after the previous CDA process. This PA process is an original path-oriented process which depends on the original path of the user's avatar. In the previous process, the system keeps the avatar's original position data and counts the number of generated right- or left- and forward- movement inputs during the Collision Avoidance process. If the CDA process does not receive any more collision detection events, then the 2D Map-Based Navigation starts the Path Adjustment process. The PA process uses the counted number of the right- or left- and forward- movements, which was saved at the second step in the previous process. In this process, it generates left- or right- and forward-moving events to make the adjusting path follow an ellipse (see Fig. 2 and 3). The left- or right-moving events offset the generated counts of the right- or left- moving events in the previous Collision Avoidance step.

4. Results and Discussion

We developed a template Multi-user 3D Virtual Reality system with JAVA and the commercial CosmoPlayer browser. We created a virtual world; it has the boundary of 1000 * 1000 and the building of 160 * 60 with a cross shape corridor in the center. Moreover, it is composed of 300 virtual objects including 12 objects inside the building (see Fig. 5). The system provides both of the navigation methods; a typical navigation method based on the use of a mouse and keyboard and the MBN navigation method, which has been proposed in this paper. Therefore the participants can also travel with the simultaneous use of typical navigation and MBN navigation.

40 students took part in the experiment study; they came from the Computer Science Dept. and the Mechanical Engineering Dept. in Yonsei University. In the post-experiment questionnaire, five questions were asked in total. First three questions covered the following three aspects of



```
AvatarURL: http //multimedia yonsei.ac.kr/~ wonlw/avatar/a
Chat Client is starting. .
Current User ..
not yet implemented
kk: hello
```

Fig. 5 MBN 3D Multi-user VR System

navigation: general movement - how simple or complicated it was to move around; placement - how difficult it was to get from one place to another; and how "natural" the movement was. From question 5 to 8, it asks question about the user has any experience with VR or not in advance, and it also gained the total time which the user used this system and MBN function. And the last two questions covered the effects of the research and the application area for this navigation interface. The questions give extra relative points to the MBN navigation method compared with the typical navigation method, which uses a mouse and keyboard. The questions and results are summazed in table 1.

Making a comparison between the users who have experience with VR and who have not (see Table 2), the users who have not experienced with VR (16 participants), gave more affirmative (8 ~ 15%) answer to the whole of the questions. The reason the most of experienced participants gave little faithless answer is this system still has

Table 1 Questions and Results of the Experiment

General Navigation	**Getting from Place to Place**	**Natural / Unnatural**
Did you find it relatively ***simple*** or relatively ***cmplicated*** to move through the computer-generated worlds?	How ***difficult*** or ***straight-forward*** was it for you to get from a place to place?	The act of moving from a place to place in the computer generated world can seem to be relatively ***natural*** or relatively ***unnatural***?
To move through the worlds was ...	*To get from a place to place was ...*	*The act of moving from a place to place seem to me to be performed ...*
1. Very Complicated	1. Very Difficult	1. Very Unnatural
10. Very Simple	10. Very Straightforward	10. Very Natural
Results	**Results**	**Results**
Mean: 8.2	Mean: 8.7	Mean: 8.971
SD.: 1.01605	SD.: 0.87615	SD.: 0.98476

Effects of Interface	**Application Area**
Do you think this (MBN) navigation interface is really helpful for the virtual environment navigation?	Which application area do you think is really good example for this interface?
For the navigation in VE, this interface is ...	I think this interface is really good for ... area.
1. Not Useful	**Results**
	1. Virtual space tour (26)
10. Very Useful	2. Virtual City (20)
	3. Shopping Mall (19)
Results	4. 3D Game (16)
Mean: 8.8	
SD.: 0.84192	

network problem so, the update rate is a little slow. Among the experienced participants, according to the period the participants used the VR systems they gave a little different results to the questions. The subjects, who have experienced with VR less than 20 min, gave the average point from 8.4 to 9.4 to the 4 questions. But the subjects, who have experienced more than 1 hour, gave the average point from 7.45 to 8.15 which is 10% ~ 14% lower than the subjects, who have experienced less than 20 min. The subjects, who have experienced less than 1 hour, also gave the average point from 7.66 to 8.94 which is 3% ~ 7% lower than the subjects, who have experienced less than 20 min. but is 2% ~ 9% bigger than the subjects, who have experienced more than 1 hour.

According to the total time of using this system, the subjects gave little different results to the questions (see Table 3). The users (3 subjects) who travel this VE less than 20 min, gave 7.0, 7.5, 7.5, and 7.5 average points to the first four questions respectively. But the subjects (12 participants) who used the system between 20 and 30 min, gave average point between 8.2 and 8.86 which is 8%~12% higher than the first group. If the subjects (16 participants) who used the system more than 40 min they gave 8.77, 8.86, 9.45, and 9.09 average point to the first four questions respectively. This results is 13%~20% higher than the first group and is also 3%~10% higher than the second and the third group for the each questions.

Table 2 Results between Experienced and Unexperienced

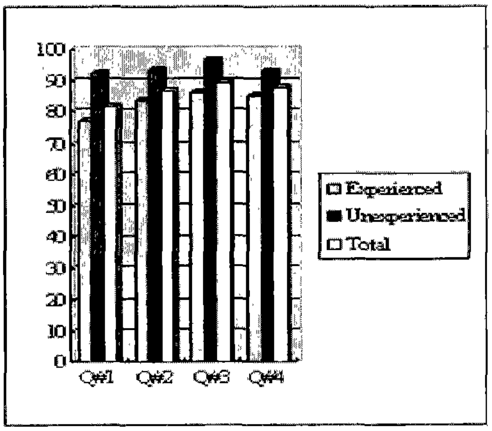
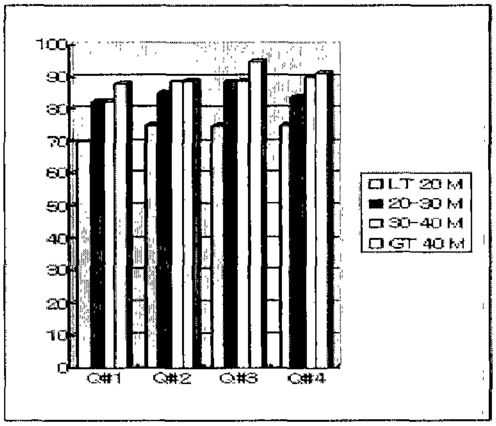


Table 3 Results according to Total System using Time



From this experiment, we can gain some conclusion, the first one is this MBN function is much useful to the user who have not experienced with VR in advance. The second one is the MBN function is a useful and helpful interface for the VR user who want to travel quite a long time in 3D VE. Even this system have still low rate network problem and limitation for multi-user VR system we can estimate this MBN navigation interface is very useful and efficient to the 3D VR users.

5. Conclusion and Future Work

The 2D Map-Based Navigation composed of those three major processes mentioned above, implements the navigation metaphor by providing more user-centered and natural navigation methods in interactive virtual environments. It also gives more useful and effective navigation services to the user, by reducing spatial loss and solving unrealistic navigation situations, like when an avatar is stalled after a collision with virtual objects or when avatars go through each other after colliding with one another. Those situations happen in the present virtual reality service systems. The MBN can improve the user's virtual Presence and Reality, which is the ultimate goal of virtual reality.

One of the effects we expect to gain from the research is a reduction in user's spatial loss in 3D VE by improving spatial awareness. When the user travels in 3D VE, sometimes the user is faced with spatial loss, one of the major problems in a virtual reality system. In order to reduce this situation, the Virtual Reality system developers must consider the user's spatial awareness. Environmental enhancements such as signs, grid-lines, and directional arrows probably will enhance navigation performance in electronic worlds. It may also be able to enhance one's spatial awareness by advancing toward more human and ecologically proper integration of multimodal stimulation and natural interaction, which will enable us to travel in virtual worlds just as well as we can in the real-world. The MBN will improve navigation performance in VE and spatial awareness by displaying the user's location information on the 2D map. The user can also decide navigation direction and destination in 3D VE, by selecting a relative position on the 2D map.

The second effect we expect from the research will be a reduction in user's fatigue and a support for parallel works. In most of the existing VR systems, the user has to input the moving event with various input devices (mouse, directional key, glove, joy stick, bike pedal, VMC, and so on) continuously when the user wants to travel in the VE. Thus, for a 3D VR user to chat with other users, the user has to either stop and type in the conversations using both hands, or type with one hand and enter each moving-event with a mouse or keyboard with the other. These situations are very unrealistic and difficult to manage. But the MBN will remove these difficult and unrealistic situations by providing a automatic continuous and constant velocity navigation function. The user can travel and chat with others in the VE with optimal convenience and with little effort.

The third effect we expect to gain from the research is an improvement in Presence and Reality in the VR. Navigation is one of the major interfaces in the VR; it can improve the Presence and the Reality by reducing the differences between the computer-modeled VE and the real environment. Because the MBN tries to provide a much simpler user-centered navigation method which is similar to the real life and to eliminate the unrealistic conditions and difficulties in some of the existing VR service systems, it will be an appropriate and convenient service tool. Therefore, the MBN can contribute to the improvement of the user's Presence and the Reality in the VR by changing the navigation method and providing a somewhat real lifelike travel and navigation in the VE for the users.

From the experimental study, we gained very encouraging results in the general navigation in VE, the natural navigation, and the navigation interface for the user in VE. For our future research, we will work toward developing more natural collision-detection technique and a more natural and smoother S.P (system processor) in avoiding or bypassing virtual objects and avatars. We will also continue with further research on effective interfaces between virtual avatars that can improve the virtual Presence and Reality of VEs.

참 고 문 헌

[1] H. Iwata and K. Matsuda. Haptic Walkthrough Simulator: Its design and application to studies on cognitive map. In The 2nd International Conference on Artificial Reality and Tele- existence, ICAT 92, pp.185-192, 1992.

[2] Doug A. Bowman and Larry F. Hodges. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environment. Proceeding of the symposium on Interactive 3D Graphics, pp.35-38, 1997.

[3] Doug A. Bowman, David Koller, & Larry F. Hodges. Methodology for the Evaluation of Travel Technology for Immersive Virtual Environments. Virtual Reality: Research, Development, and Applications, vol. 3(2), pp.120-131, 1998.

[4] J. Mackinlay, S. Card, and G. Robertson. Rapid Controlled Movement Through a Virtual 3D Workspace. *Proceedings of SIGGRAPH* (Dallas, TX, 1990), in *Computer Graphics*, vol. 24, no. 4,

pp.171-176, 1990.
[5] R. Pausch, T. Burnette, D. Brockway, and M. Weiblen Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures. *Proceedings of SIGGRAPH*(Los Angeles, CA), pp. 399-400, 1995.
[6] E. Strommen. Children's Use of Mouse-Based Interfaces to Control Virtual Travel. *Proceedings of CHI* (Boston, MA), pp.405-410, 1994.
[7] Stoakley, R., Conway, M., Pausch, R. Virtual Reality on a WIM: Interactive Worlds in Miniature. In proc. CHI'95, pp.265-272, 1995.
[8] M. Mine. Virtual Environment Interaction Techniques. UNC Chapel Hill Computer Science Technical Report. TR95-018, 1995.
[9] Mel Slater and Martin Usoh. Representations systems, perceptual position, and presence in immersive virtual environments. Presence, 2(3), pp.221-233, 1993.
[10] Mel Slater, Martin Usoh, and Anthony Steed. Taking steps: The influence of walking technique on presence in virtual reality. ACM Trans. On Computer-human Interaction, 2(3), pp.201-219, 1995.
[11] Darken, R. and Sibert, J. A Toolset for Navigation in Virtual Environment, Proceedings of ACM User Interface Software & Technology, pp.157-165, 1993.
[12] Darken, R. and Sibert, J. Navigating in Large Virtual Worlds. The International Journal of Human-Computer Interaction, 8(1), pp.49-72, 1996.
[13] Glenna A. Satalich. Navigation and Wayfinding in VR: Finding Proper Tools and Cues to Enhance Navigation Awareness. Master's thesis, University of Washington, 1995.
[14] Dongbo Xiao and Roger Hubbold. Navigation Guided by Artificial Force Fields, CHI'98, pp.18-23, 1998.
[15] Tsai-Yen Li, Jyh-Ming Lien, Shih-Yen Chiu, and Tzong-Hann Yu. Automatically Generating Virtual Guided Tours. Computer Animation, Proceedings, pp.99-106, 1999.

권 태 욱

1986년 3월 육군사관학교 졸업(이학사). 1995년 9월 미 해군대학원 컴퓨터공학 석사. 2001년 2월 연세대학교 컴퓨터과학과 박사. 2001년 2월 ~ 현재 육군사관학교 전산학과 순환교수. 관심분야는 멀티미디어, 가상현실(VR), HCI, Wearable Computer.

최 윤 철

1773년 서울대학교 전자공학과 졸업(공학사). 1975년 6월 공학 석사(Univ. of Pittsburgh). 1979년 6월 공학 박사(Univ. of California, Berkeley, Dept. of IE&OR). 1979년 8월 ~ 1982년 7월. Lockheed사 및 Rockwell International사 책임 연구원. 1982년 9월 ~ 1984년 1월 Univ. of Washington 전산학. 1990년 9월 ~1992년 1월 Univ. of Massachusetts 연구 교수. 1984년 ~ 현재 연세대학교 컴퓨터과학과 교수. 관심분야는 멀티미디어, 하이퍼미디어, 지리정보 시스템(GIS)