

Article

Maze Terrain Authoring System in Immersive Virtual Reality for New Visual Realism

Jinmo Kim 

Division of Computer Engineering, Hansung University, Seoul 02876, Korea;
jinmo.kim@hansung.ac.kr; Tel.: +82-2-760-4046

Received: 6 March 2019; Accepted: 1 April 2019; Published: 4 April 2019



Abstract: This paper proposes a maze terrain authoring system with which a user can automatically calculate various and complex maze patterns to compose maze terrains in an easier and more intuitive structure. Using the maze terrain information calculated by using the proposed authoring system, a 3D maze terrain is generated quickly and effectively, and through this, the user's visual realism in an immersive virtual reality is increased to provide a new presence. The proposed maze terrain authoring system consists of three core functions: a function that automatically generates a grid maze of various sizes and patterns based on a maze generation algorithm; a function that calculates a circular maze in an intuitive structure; and a function that transforms a sketch-based maze by using an image-processing algorithm. Through the authoring system consisting of these functions, various maze terrains from uniform mazes to irregular mazes can be generated effectively. This study produces maze terrains of various concepts directly by using maze information calculated through the authoring system. In addition, through experiments, this study analyzes the efficiency in an immersive virtual reality and the presence through visual realism. In this process, the suitability of the authoring system is also analyzed in combination with a survey.

Keywords: immersive virtual reality; virtual maze environment; authoring system; visual realism; presence

1. Introduction

To provide a realistic experience, virtual reality (VR) facilitates interaction that satisfies the diverse senses of a user such as visual, auditory, and tactile senses. Here, immersive VR focuses on presence, whereby a user can feel a realistic experience based on the senses regarding where he/she is, whom he/she is with, and what actions he/she is performing. To elevate the presence of a user in an immersive VR, it is important to satisfy the visual realism by transmitting 3D image information first.

For this purpose, devices such as a 3D monitor and head mounted display (HMD) are used. In addition, to increase the immersion by inducing user participation, studies have been carried out for methods that can detect actions and movements and interact more directly with the virtual environment or objects. Studies were carried out for haptic systems that not only detect the movements of human body joints accurately and express them realistically in a virtual environment but also make accurate calculations and provide feedback regarding physical processes and reactions such as force and tactile senses that occur in this process. Furthermore, studies were performed for motion platforms that implement free walking in a spacious virtual environment in a limited real environment. Lately, by using various approaches of psychology, neural science, and social phenomena, studies have been conducted to analyze the effects of VR on our lives.

In terms of increasing the presence in immersive VR, how a user interacts with the virtual environment is important, but equally important is how diversely virtual scenes are provided to visually satisfy the user. However, because hardware-oriented VR studies (haptic systems, motion

platforms, etc.) for the realistic expression of user actions and movements in immersive VR considered only the scenes for experiments, the space was limited, and the structure was simple in most cases. Despite this, because the VR technology is applied to various areas such as fine arts and architectural design as well as games, specialized studies are needed to create a complex virtual space in accordance with the VR. To this end, studies have been carried out for the methods [1–3] of using procedural modeling to automatically create virtual scenes such as urban spaces and virtual landscapes, which have complex structures and configurations and distinct characteristics. However, few studies have been conducted on creating novel scenes for the users' visual realism in immersive VR, or on generating terrains suitable for the environment for experiencing VR.

Accordingly, this study aims to develop immersive VR based on visual realism that provides users with novel presence sensations and experiences. For this purpose, we conduct novel research on a technique for creating maze terrains and develop a new design for an authoring system to support the technique. This system is user-oriented and includes various functions that calculate diverse maze patterns automatically and generate terrains in an easy and intuitive structure. The proposed maze terrain authoring system consists of the following core functions:

1. A function is designed to create and edit a maze terrain the way a user wants in an easy and intuitive structure by automatically calculating the maze pattern, in which a path from the entrance to the exit is clear, by using a maze creation algorithm.
2. For the diversity of maze patterns, a circular maze generation function is designed that generates a circular maze automatically by using simple parameters that can be easily understood by the user.
3. A sketch-based maze transformation function is designed that automatically converts a hand-drawn maze sketch image of the user into a 3D maze terrain through a morphological image-processing operation.

Based on these three core functions, a user can easily and intuitively create desired 3D terrains that have various irregular maze patterns.

2. Related Works

2.1. Immersive VR

Studies on immersive VR have been carried out using various approaches: a display that transmits 3D visual information so realistic experiences can be provided to a user in a virtual environment based on the five senses, sound processing for providing improved spatial perception by using a volumetric audio source, a method of interacting with the virtual environment more directly, and a haptic system and motion platform that feed physical reactions back to the human body (hand, foot, etc.).

To provide an improved presence to users through high immersion, it is necessary to provide interactions that are so realistic that the boundary between VR and reality is vague [4,5]. To this end, some studies have been carried out to accurately detect the movements of human body joints moving in a real space and to analyze the intention of the movements or the process of action to reflect it in the virtual environment.

By attaching a surface or optical marker to a hand joint and detecting movements based on the marker, many studies including that of Matcalf et al. [6] have proposed methods of mapping the detecting movements to the motions of a virtual hand model [7,8]. Recently, techniques that express realistic movements in virtual spaces have been developed, e.g., a sphere-mesh tracking model proposed by Remelli et al. [9] and Total Capture (hand gestures, facial expressions, and whole-body movements are captured) proposed by Joo et al. [10].

In addition, to express infinite walking in a limited indoor space, studies were carried out on smooth assembled mapping [11] and a walk-in-place detection-based simulator [12]. Moreover,

at present, methods have been developed to express the natural walking of multiple users in the same space [13].

Presence-oriented immersive VR studies are making progress in proposing novel ideas related to haptic systems that feedback physical reactions occurring in an interaction process between the user and the virtual environment. Related studies include those on 3-RSR (Revolute-Spherical-Revolute) [14] or 3-DoF (Degrees of Freedom) [15] wearable haptic devices and portable hand haptic systems [16]. These devices handle tactile senses delicately and improve the immersion of the user in VR, thereby providing realistic experiences.

Studies on presence based on that of Slater et al. [17] have been carried out by using various cases. The user experience cases of VR have been analyzed from various perspectives such as walking [18] and communication between users [19]. Moreover, these studies made headway in terms of comprehensively analyzing the convergence with other fields of study such as psychology and neural science, in addition to the psychological effects of the VR environment and conditions on user behavior and perception [20,21].

Furthermore, studies have been carried out to analyze the factors that improve the presence in interactions using gazes and hands in the application aspect of immersive VR [4,5,22]. To satisfy the improved presence of the user in immersive VR, visual satisfaction should be preceded. For this reason, interest is growing in multimodality studies [23,24] that consider visual and auditory senses or visual and tactile senses simultaneously, and studies on pseudo-haptics that combine visual experiences and haptic systems [25,26]. As such, various approaches have been made to enhance presence sensation in immersive VR. However, there are relatively few studies on scene generation methods that focus on the visual realism of users in immersive VR to provide new and diverse experiences.

2.2. Virtual Scene for Visual Realism

A virtual scene generation method that can provide an improved presence along with new visual experiences to a user in a virtual environment is a required technique for the diversity of immersive VR applications. In the future, VR scenes will break free from small and limited spaces such as indoor spaces and create vast outdoor terrains such as cities and landscapes in virtual environments so that a greater scope can be provided to users.

Through the procedural modeling of computer graphics [27], shape grammar [28], and image-based modeling [29], Vanegas et al. [1] defined the collection of buildings, parcels, etc., and surveyed many studies that efficiently generated and visualized urban spaces composed of streets and roads. Dylla et al. [30] reconstructed a virtual city by setting Rome as a specific region. However, to prevent side effects such as VR motion sickness, immersive virtual scenes must consider the rendering performance (refresh rate, frames per second (fps), etc.) in addition to realistic expressions.

Considering this point, Kim [2] improved virtual landscape generation techniques [31,32] and proposed an optimized modeling and optimization method for immersive VR. Jeong et al. [33,34] conducted studies on a maze generation method for new scene generation in an immersive VR.

With the goal of generating virtual scenes of mazes that can provide new experiences and an improved presence to users in immersive VR by increasing the satisfaction of the visual experience, this study builds on conventional studies and proposes an optimized authoring system that can generate more diverse virtual scenes of mazes in an easy, intuitive, and user-oriented structure.

3. Maze Terrain Authoring System

The maze terrain authoring system of this study generates a maze-based immersive virtual terrain while aiming to increase the presence of visual realism to users in immersive VR through various visual experiences. To this end, three core functions are designed that calculate various maze patterns automatically in an easy, intuitive, and user-oriented structure, and convert them to information required for 3D maze terrain generation.

First, a maze-generation-algorithm-based maze pattern generation function is defined whereby mazes of various sizes and patterns are automatically calculated and can be edited freely by the user. Next, a circular maze generation function is designed to escape the monotony of rectangular maze patterns. Lastly, a sketch-based maze transformation function is implemented to effectively convert irregular maze patterns into a maze terrain.

Figure 1 shows the proposed maze terrain authoring system. The user interface consists of the menu at the top, the 2D maze pattern information viewer on the lower left side, and the 3D maze terrain viewer on the lower right side. Based on the algorithm and calculation processes proposed in this study, the system is configured to enable the user to effectively input and use the required parameters. The proposed authoring system was designed with a function to generate maze terrains using a height map, enabling users to control the resolution of 3D maze terrain and the number of polygons. Thus, the developer can effectively control conditions such as the refresh rate and fps, which are required to create large maze terrains in immersive VR.

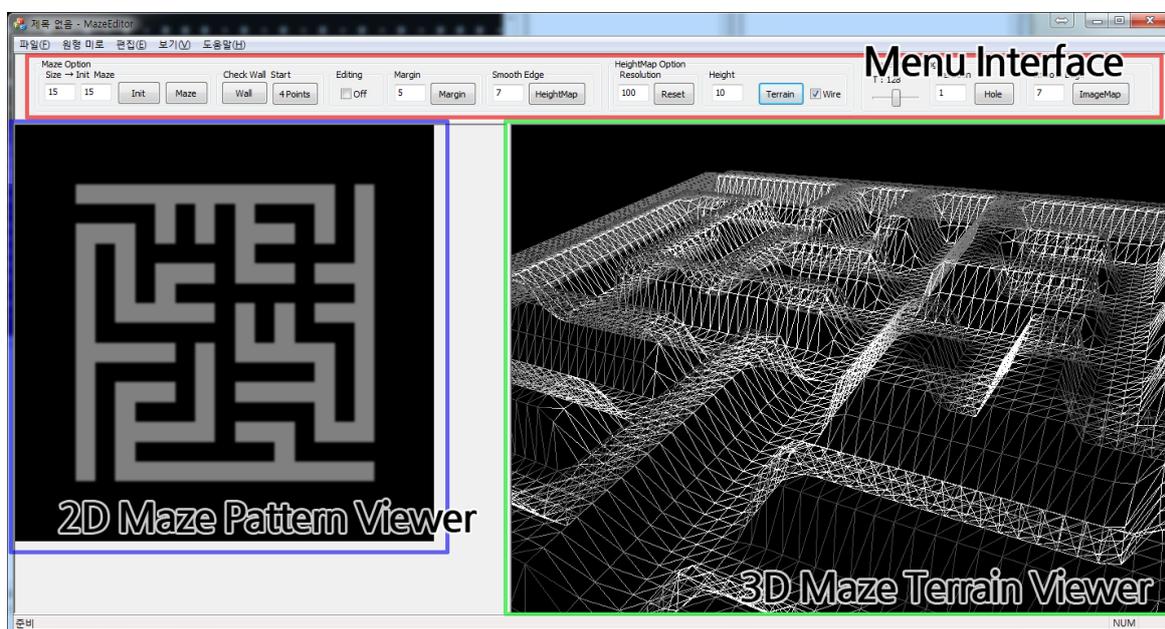


Figure 1. User interface of the proposed maze terrain authoring system.

3.1. Maze Pattern Based on the Maze Generation Algorithm

The first function of the maze terrain authoring system is to automatically generate maze patterns by using the maze generation algorithm. An important factor when generating a maze pattern automatically is that a finite maze must be generated. Various paths or one absolute path should exist so that a user is not trapped inside the maze. This helps to prevent users from experiencing VR sickness, a prominent issue in immersive VR. This is caused by feelings of frustration and anxiety induced in a closed space, such as when trapped in a maze. This study designs the function of the maze terrain authoring system based on an algorithm of Lee et al. [22] that calculates the maze pattern by using Prim's algorithm [35].

The finite maze generation algorithm of Lee et al. [22] is a method that starts with the processes of (1) setting a starting position and (2) randomly setting a road cell from surrounding cells. Then, (3) the algorithm examines the four surrounding directions (defining the layer structure of the cells after registering the four surrounding direction cells of a road cell as a wall cell set). The algorithm then repeats the processes of (4, 5) to find one among the registered wall cells, and then creates a road.

When the wall and road cells are determined through the processes of the five steps, a process is performed to calculate a 2D maze texture map that is used as the height map of the 3D maze terrain. This study composes the menu of the authoring system by classifying only the values that need to be adjusted through inputs of the user during the processes from maze pattern calculation to maze terrain information generation.

Figure 2 shows the result of composing the menu of the authoring system with the parameters needed for the automatic generation of maze patterns based on the maze generation algorithm. The user sets a maze pattern of a wanted size, and when the button Maze is pressed, the maze pattern is automatically calculated based on the algorithm of Lee et al. [22]. Moreover, a button named 4Points is provided so that the entrance, which is needed by the user to experience the maze terrain, can be connected to the roads at the ends of the four directions.

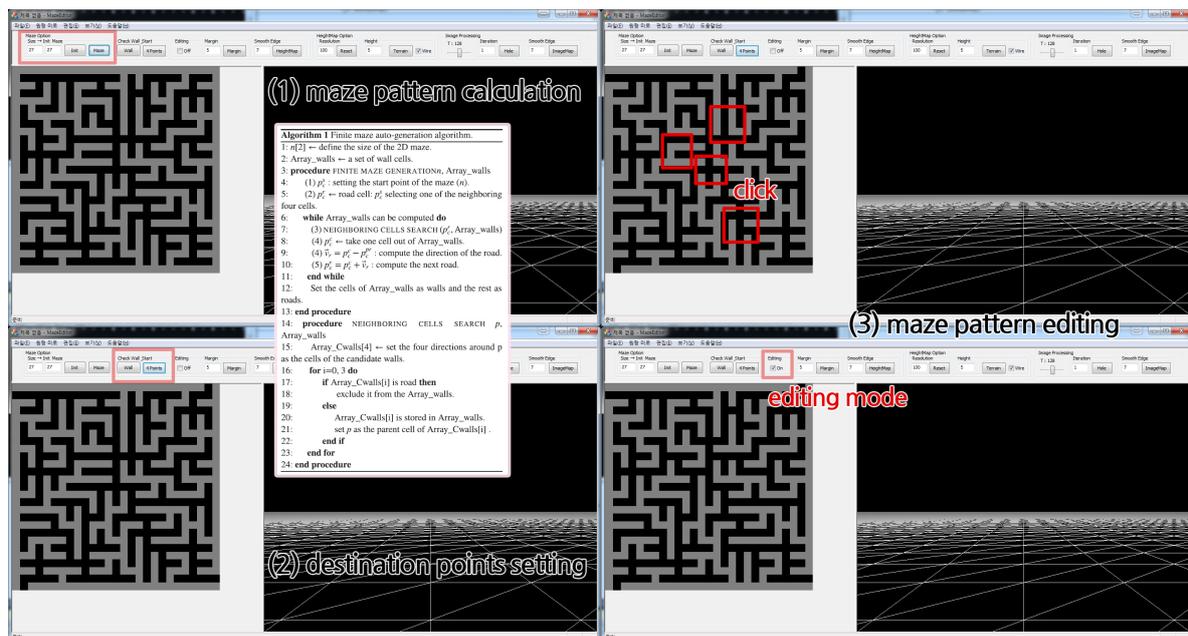


Figure 2. Maze pattern calculation process based on the maze terrain authoring system (Calculation of a cell maze pattern using a finite maze-generating algorithm developed by Lee et al. [22] → Click the destination points → Click to edit the maze terrain of cell (road or wall)).

Thus, far, we have shown the process of automatically and diversely generating a maze pattern of a size wanted by the user. Here, when the user wants to edit the automatic maze pattern, the Editing checkbox is activated. Afterwards, the user moves the mouse to directly click and edit. When the mouse is clicked directly on the maze pattern information view of the authoring system, the cell information is set to automatically convert a wall to a road and a road to a wall. Based on this, Figure 3 shows the process of generating a 2D maze texture map for 3D maze terrain generation. By inputting a margin value for the surrounding scenery setting of the maze and a blur value (smooth edge) that controls the smoothness of the maze wall, the user extracts a 2D maze texture map of $2n \times 2n$ size that is required for 3D maze terrain generation.

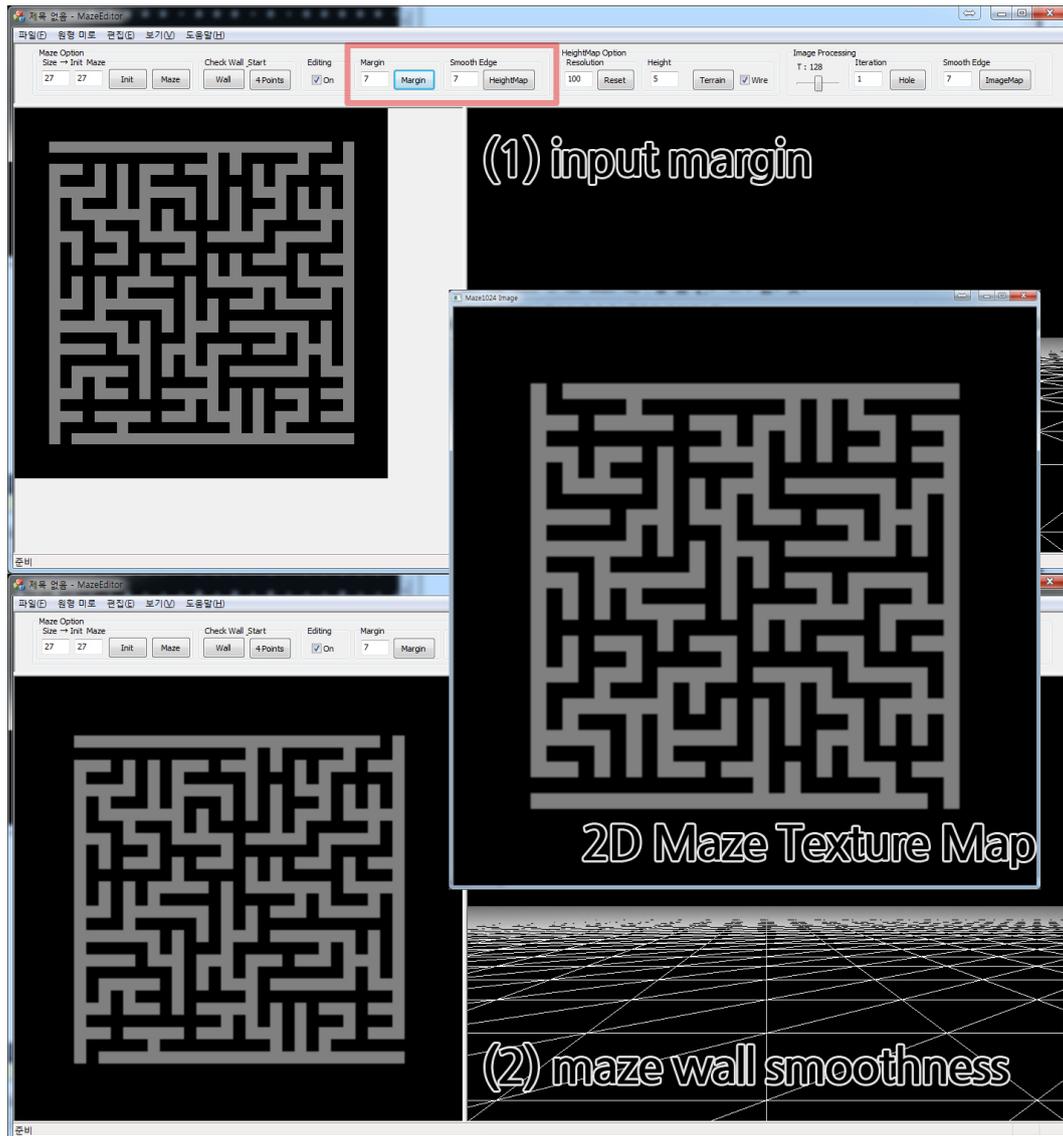


Figure 3. Two-dimensional maze texture map-producing process for 3D maze terrain generation (Input a margin value for the background settings → apply blurry image processing to create smooth wall corners).

When a 2D maze texture map is generated, the maze terrain view of the authoring system lets the user see the maze pattern in 3D through the 3D maze generation. Here, the resolution of the 3D maze terrain and the height of the terrain based on the height map can be controlled arbitrarily by the user to check the maze pattern. If the terrain function of a game engine such as Unity 3D is used, the maze terrain can be checked without a separate modification process through the generated height map.

Figure 4 illustrates the process of checking the maze generated through the maze terrain view of the proposed authoring system. Furthermore, it is also confirmed that the maze terrain can be checked through the maze texture map produced by a commercial engine such as Unity 3D.

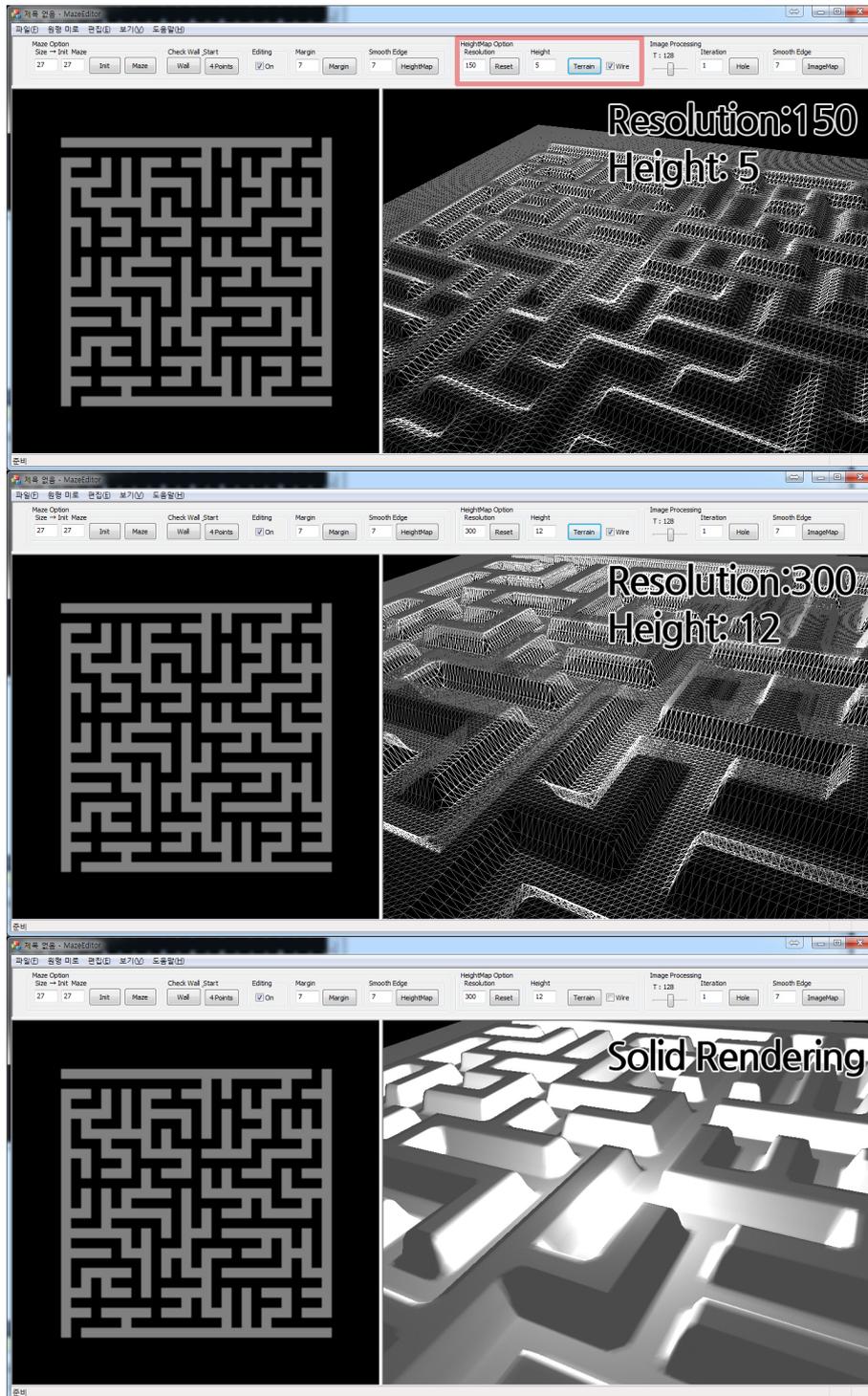


Figure 4. Visualization of 3D maze terrain through the viewer function of the proposed authoring system.

3.2. Circular Maze Generation Method

The second function of the proposed maze terrain authoring system is a circular maze generation method. This method automatically generates circular mazes of various sizes and patterns in addition to providing the maze pattern generation function of a rectangular grid type. Thus, this method expands the users' visual experiences in immersive VR. The process of distinguishing and setting up a

wall and road cells is basically identical. However, according to the characteristics of the circular maze pattern, the radius of a circular maze and the number of maze entrances are set up as input information.

First, when a user inputs a desired maze radius, the wall cells of the circular maze are automatically calculated by using a midpoint circle drawing algorithm. Then, a wall of the circular maze is repeatedly calculated at a certain interval (n_s) from the outer circle. Next, based on the number of entrances inputted by the user, the passages are created by rotating the wall cells as many times as the number of entrances from the outermost circle.

The cells that will become the entrance beginnings in the outermost circle are selected arbitrarily as many times as the number of entrances. Then, in the selected cell, a wall cell is selected whereby the length is the same as the interval (n_s) between the circular walls, and it is rotation-transformed just enough to touch the inner circular wall. Lastly, the wall cell of the position where the rotation-transformed wall cell touches the inner circular wall is found, and the above processes are repeated.

Algorithm 1 shows the process of automatically generating a circular maze pattern based on the circular maze radius and number of entrances inputted by a user. Figure 5 shows the process of generating a circular maze by using the authoring system shown in Algorithm 1. The processes of generating a maze texture map from the generated circular maze pattern and visualizing the maze terrain are performed in the same manner as the content of Section 3.1.

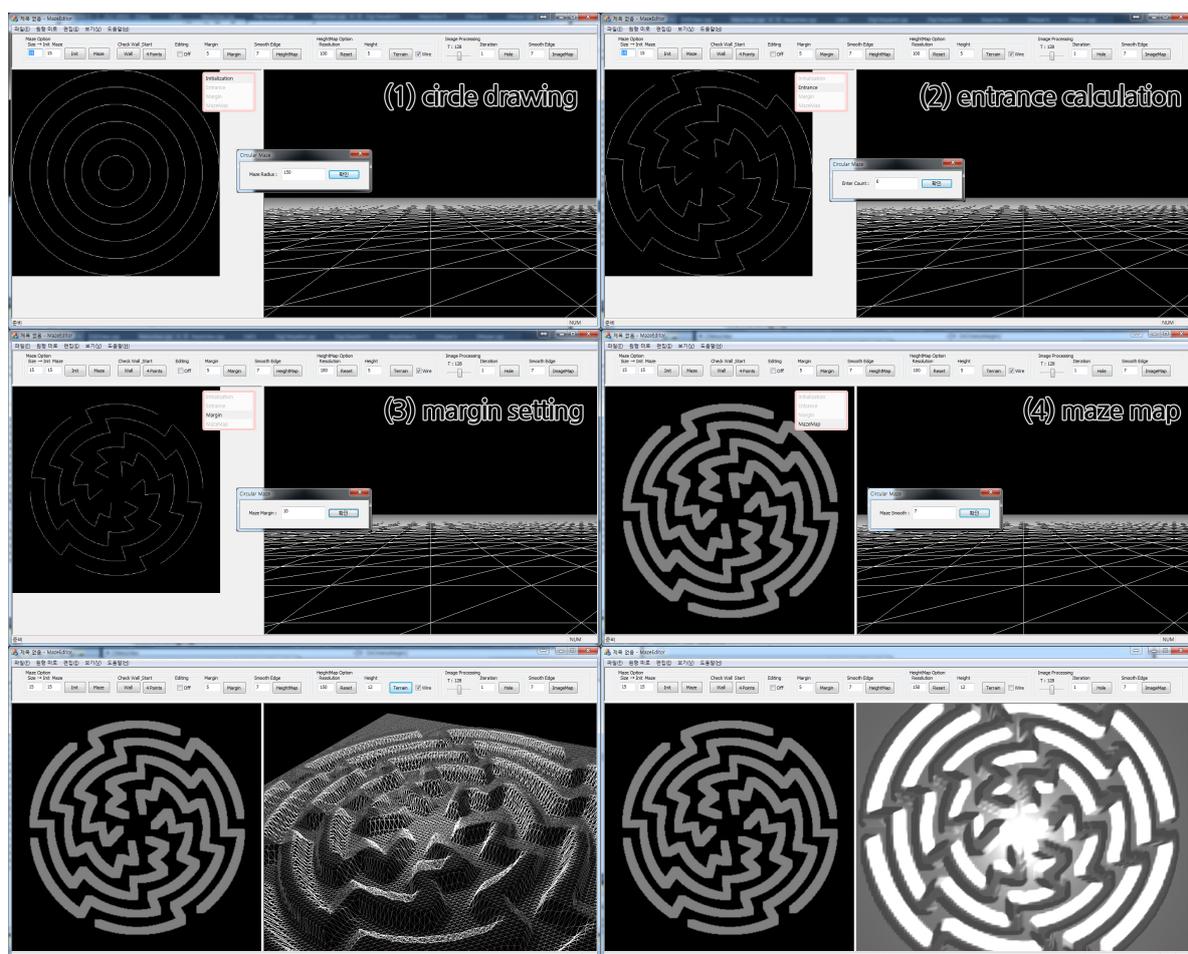


Figure 5. Circular maze pattern generation process through the maze terrain authoring system.

Algorithm 1 Circular maze generation algorithm design.

```

1:  $n_r \leftarrow$  radius of the circular maze.
2:  $n_e \leftarrow$  number of maze entrances.
3: procedure CIRCULAR MAZE GENERATION PROCESS( $n_r, n_e$ )
4:    $n_s \leftarrow$  interval between maze for user road.
5:   while  $n_r > n_s$  do
6:     define wall cell by using Bresenham's circle drawing algorithm.
7:      $n_r = n_r - n_s$ .
8:   end while
9:    $n_c \leftarrow$  number of drawing circles.
10:   $Arr_e[n_e] \leftarrow$  as many entrance cells as  $n_e$  are selected from wall cell of the outermost circle.
11:  for  $i = n_c, 0$  do: generate passages in every circle.
12:    for  $j = 0, n_e$  do
13:      select cells with length of  $n_s$  in entrance cell ( $Arr_e[j]$ ).
14:      rotate selected cells.
15:      renew  $Arr_e[j]$  by finding a cell that meets the inner circle after rotation.
16:    end for
17:  end for
18: end procedure

```

3.3. Sketch-Based Maze Transform

The last function of the maze terrain authoring system of this study is sketch-based maze transformation. The basic premise of the maze pattern in Section 3.1 and circular maze in Section 3.2 is how a uniform maze is generated. However, in many cases, a maze park or maze theme park that we can experience in daily life has an irregular shape. Therefore, this study designs a function that involves a user's hand-drawn sketch image so that a natural maze terrain can be easily generated as well as a maze pattern of artificial feeling. The last function enables maze terrains to be used in various applications, such as systems for finding directions or virtual parks.

The proposed transformation function enables a user to draw a maze pattern by using graphic software such as Photoshop or by scanning a sketch that is hand-drawn on a piece of paper. Next, the saved image is loaded into the authoring system, and by applying the image-processing algorithm, it is converted into a maze texture map. The process is carried out in three steps.

The first step is a process of fetching the sketch image in the authoring system. For the generalized computations of subsequent processes, the sketch image is transformed into a one-channel grayscale image. The second step is a process of classifying the walls and roads through binarization. Here, a threshold value is an important element for the classification of walls and roads through binarization. There is a method of finding and applying an adoptive threshold value automatically according to the input image, but because environments for producing input images are all different, the process of classifying walls and roads cannot be accurately performed.

Therefore, this study provides a threshold value through the interface of the authoring system so that the user can intuitively control it. The third step is a process of filling holes that occur in the binarization process. Through a morphology close operation, the holes occurring in the walls are filled. Consequently, a natural maze is generated. Here, a system menu item is provided to enable the user to control the number of iterations of the morphology operation.

Figure 6 shows the process of generating a maze texture map in steps from the user's hand-drawn maze sketch through the sketch-based maze transformation function. After the third step, the maze texture map production process through the margins and smooth settings is the same as that discussed earlier.

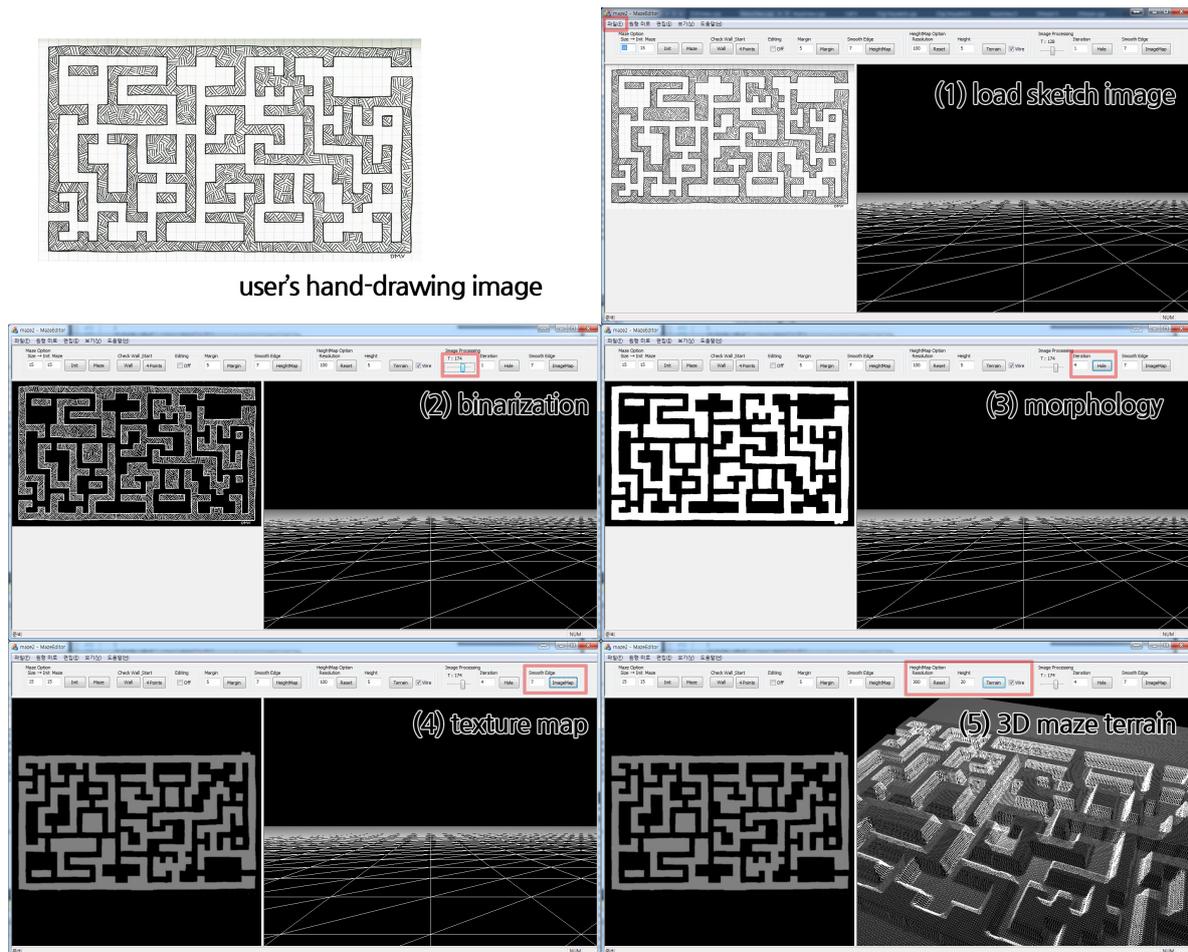


Figure 6. Process of the sketch-based maze transformation function.

4. Experimental Results and Analysis

The proposed maze terrain authoring system was implemented using Microsoft Visual Studio 2013, OpenCV 2.4.11, and DirectX SDK 9.0c. Furthermore, a Unity 2017.3.1f1 (64-bit) engine was used to generate a maze terrain finished through graphic work based on a maze texture map generated automatically by using the maze authoring system. To test the maze terrain confirmation and experience the result in immersive VR, the system was implemented by integrating the Oculus Rift CV1 HMD device and Oculus SDK (ovr_unity_utilities 1.22.0) with the Unity3D engine. Finally, the PC used for the experiment and implementation had an Intel Core i7-6700, 16 GB of RAM (Random Access Memory), and a Geforce 1080 GPU (Graphics Processing Unit).

First, the process of generating a maze terrain through the maze terrain authoring system is examined. Figure 7 shows this, and a total of three virtual maze terrain scenes are composed. These are the results of generating terrains from the maze patterns calculated based on the core functions of the authoring system, i.e., the maze algorithm-based maze pattern function, circular maze generation function, and sketch-based maze transformation function. Furthermore, it can be seen that variously directed virtual maze environments are constructed by adding graphical factors according to the concept.

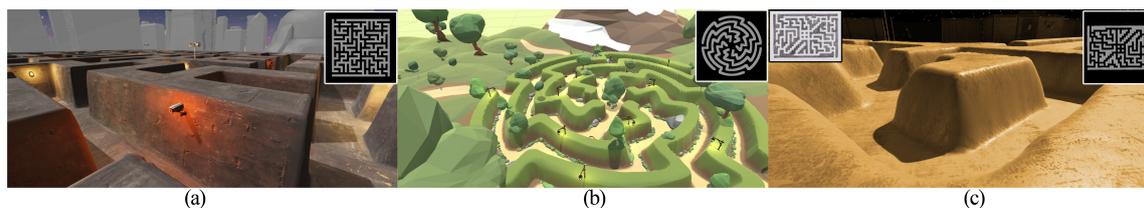


Figure 7. Results of generating maze terrains using various concepts by adding graphical factors: (a) maze pattern function (deserted city), (b) circular maze function (casual land), and (c) sketch-based maze transform function (mars terrain).

Next, an analysis is performed for the basic performance that is necessary when a user experiences a generated maze terrain without VR motion sickness in VR. In general, for scenes of VR content, rendering must be performed for two images transmitted binocularly, unlike in a game. Therefore, to minimize the occurrence of dizziness in the process of scene transition, it is recommended that 75 frames per second should not be exceeded, and the total number of polygons should not exceed 2 M in a rendering scene. Table 1 lists the comparison results between a basic maze terrain that has no graphical factor and maze terrains that contain graphical factors by measuring the number of frames per second. The measurements were taken by changing the camera position 10 times in total, and the average was recorded.

Table 1. Technical analysis result of maze terrain generated based on the proposed authoring system.

	Frames Per Second (fps)		
	Minimum	Maximum	Mean
Non-Graphical Factors	99.21	130.40	117.15
Deserted City	78.89	89.18	86.78
Casual Land	90.19	118.27	101.20
Space Mars	81.28	92.29	88.28

Here, a difference occurred in the number of polygons according to the graphical factors, and this can have a direct impact on the number of frames per second. Therefore, after measuring the performance in a state of excluding the graphical factors, the differences were compared after placing the graphical factors. Considering the VR motion sickness of the user, the three virtual maze scenes all maintained several polygons under 2 M.

For the deserted city, 0.248 M minimum and 1.811 M maximum were recorded, and here, the number of frames per second was 86.78 on average, showing a difference of 30.37 from the basic maze terrain. The casual land scene that showed the smallest number of polygons (0.101 M minimum and 0.612 M maximum) showed a difference of 15.95 from the basic maze. Therefore, it is confirmed that if the specifications for graphical factors in a maze terrain are carefully considered, they are suitable to be applied as VR content in terms of the overall technical conditions.

Lastly, experiments were performed based on surveys to check whether the maze terrains generated by using the proposed authoring system improved the presence through the high visual immersion of the user in VR. A total of 15 survey participants of ages 21–37 were randomly selected, and while wearing the Oculus Rift CV1 HMD, they experienced the provided maze environment. Figure 8 shows the experience environment of this study. Moving of virtual characters on the maze terrains was implemented through key inputs of Oculus Touch.

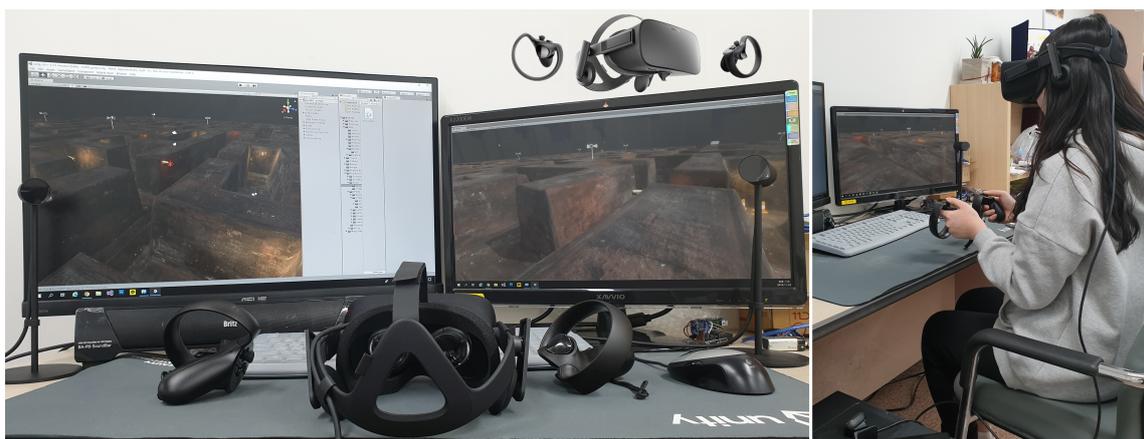


Figure 8. Experience environment for the immersive maze virtual environment.

The first survey experiment was performed to analyze the efficiency and satisfaction of the system from the aspect of the users (developers) that used the proposed authoring system. Through the survey, it was checked if the proposed authoring system processed the maze terrain generation efficiently and conveniently from the perspective of developers or level designers who produce VR scenes. The survey was conducted by randomly selecting six people: two persons each from software developers, designers, and students majoring in software.

By dividing the three functions of the authoring system, the time taken to produce a wanted maze and satisfaction value (1–5) for respective functions were recorded. Table 2 lists these results. For an objective comparison of the proposed authoring system, the time consumed for producing the maze terrains by using the terrain tool provided by the Unity 3D engine was also recorded.

First, in the case of the maze pattern calculation function, the editing times were considered together, and 47.17 s was consumed on average. In the second circular maze case, because no separate editing function was provided, the shortest time was consumed with an average of 24.35 s. The last sketch-based maze transformation function took the longest time with a 192.77 s average as a result of taking the measurement by considering the time taken for a user to draw the maze pattern by hand. According to the existence/nonexistence of experience in hand drawing and using graphical tools such as Adobe Photoshop, large time deviations occurred.

Finally, as a result of measuring the time taken to produce a wanted maze terrain by using the terrain tool of the Unity 3D engine, 1029.81 s was consumed on average. The six users who participated in the experiment had experience with the Unity 3D engine at least once. When the maze terrains were generated through the terrain tool of the engine under these conditions, because the maze patterns had to be planned and fabricated manually by using brushes, a significant amount of time was required.

In the case of satisfaction, the highest satisfaction was shown in the first maze pattern function that provided the generation function of various maze patterns and the editing function to the user at the same time. The second highest satisfaction was shown in the sketch-based maze transformation function, which automatically transforms a hand-drawn maze image, although it takes longer. In the case of the circular maze, because there were fewer functions for the user to select and the results were monotonous, the satisfaction was low.

Table 2. Survey analysis results for usability and efficiency of the proposed authoring system.

Factors	Average Time (SD)	Satisfaction (SD)
1. Grid Maze Pattern	47.17 s (0.815)	4.252 (0.258)
2. Circular Maze Pattern	24.35 s (0.254)	3.579 (0.344)
3. Sketch-Based Maze Transform	192.77 s (7.911)	3.978 (0.277)
4. Terrain Function of Unity 3D	1029.81 s (15.553)	-

Lastly, a survey analyzes the user's visual realism. The survey experiment was performed with the goal of checking if the virtual maze environment produced through the proposed authoring system provided satisfactory visual realism to the users. From questions proposed by Witmer et al. [36], this study used four questions (4, 6, 7, and 13) related to "visual" in the realism category and recorded the values.

Table 3 lists the results of recording the survey values of users. After experiencing all three maze terrains, the users recorded the experience in each scene with a value between 1 (NOT AT ALL) and 7 (COMPLETELY). As a result, it was confirmed that a majority showed high visual satisfaction close to 6. To check whether the graphical factors of different concepts affected the experience on the terrain, the statistical significance was calculated through a one-way ANOVA (analysis of variance). As a result, $F(2,42) = 2.0712$, and the p-value was 1.456×10^{-1} , confirming that there was no significant difference between the three scenes in terms of visual realism.

Table 3. Statistical analysis results for visual realism provided by maze terrain in immersive VR.

Visual Realism	Mean	SD
Deserted City	6.223	0.311
Casual Land	5.996	0.257
Space Mars	6.077	0.293
Pairwise Comparison		
All Functions	$F(2,42) = 2.0712, p > 0.05, p = 1.456 \times 10^{-1}$	

5. Limitation and Discussion

This study produced maze terrains of various concepts using the three functions of the proposed authoring system and conducted tests to examine system performance required in immersive VR and verify their application feasibility. In addition, a questionnaire survey was conducted to identify whether developers were sufficiently satisfied with the authoring system and could easily and intuitively use it. This study, however, found difficulty in conducting a test to directly compare the proposed system with typical terrain-generating authoring systems or relevant software. A comparative test was conducted with the terrain-generating function of a Unity 3D engine, which supports immersive VR applications, but comparisons with more diverse studies are needed for credible verification.

Current maze terrains can be used as a background in the fields of entertainment and tourism, such as in games or maze parks, respectively. These maze terrains can be further developed into serious games that train or analyze users' sense of space or direction, or analyze or verify the performance of systems for finding directions. However, additional research is needed for functions matching the purpose of various other areas of application.

6. Conclusions

The maze terrain authoring system proposed in this study aimed to provide new presence by increasing the immersion of the user in visual realism in immersive VR. Accordingly, the system was proposed to efficiently generate various maze terrains in a user-oriented intuitive structure. To this end, the authoring system was designed by classifying three functions.

Firstly, a maze-generation-algorithm-based maze pattern generation function was implemented in the authoring system to automatically calculate the maze patterns of various sizes when generating maze terrains of a grid type and to edit them in a desired shape easily and intuitively. To express a circular maze terrain, a circular maze generation algorithm was defined and provided as a function of the authoring system. Lastly, a sketch-based maze transformation function was designed so that irregular maze terrains could be expressed in contrast to the uniform feeling provided by maze patterns generated by the algorithms.

In this method, a user draws a desired maze pattern by hand, and the information required to generate the maze terrain is calculated automatically through the image-processing algorithm. The authoring system was completed by implementing a viewer that can check the 3D maze terrain in real time based on the maze pattern calculated by the authoring system equipped with the three functions. From the maze texture map generated through the proposed authoring system, the maze terrain was produced directly, which satisfies the technical performance required by VR applications, to analyze the visual realism of users in immersive VR. VR applications require better performance (more than 75 fps) than interactive systems such as games because of delay, which may occur when generating scenes and trigger VR sickness. In this respect, the maze terrain produced in this study was verified via tests to be suitable for use as a VR application.

Moreover, a survey experiment was conducted by targeting general participants. Firstly, as for the system performance, through comparative tests with the terrain function of commercial engines, it was confirmed that the proposed authoring system enabled users (developers) to offer desired maze terrains easily and intuitively. This shortened required time and led to high user satisfaction with the system. Lastly, through the realism category of a survey questionnaire about presence sensation, it was confirmed that maze terrains in immersive VR could offer novel visual experiences to users. It was also verified through a statistical analysis that all terrains produced through the proposed three functions were satisfactory with no significant differences between them, which may improve presence sensation. In the future, the system will be further developed to provide a large-scale VR maze experience environment such as a maze park or theme park, where multiple users can experience the maze space together in VR.

Funding: This research was financially supported by Hansung University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vanegas, C.A.; Garcia-Dorado, I.; Aliaga, D.G.; Benes, B.; Waddell, P. Inverse Design of Urban Procedural Models. *ACM Trans. Graph.* **2012**, *31*, 1–11. [[CrossRef](#)]
2. Kim, J. Modeling and Optimization of a Tree Based on Virtual Reality for Immersive Virtual Landscape Generation. *Symmetry* **2016**, *8*, 93. [[CrossRef](#)]
3. Jung, S.; Kim, D.; Kim, J. GeoMaTree: Geometric and Mathematical Model Based Digital Tree Authoring System. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 3284–3306.
4. Kim, M.; Lee, J.; Jeon, C.; Kim, J. A Study on Interaction of Gaze Pointer-Based User Interface in Mobile Virtual Reality Environment. *Symmetry* **2017**, *9*, 189. [[CrossRef](#)]
5. Han, S.; Kim, J. A Study on Immersion of Hand Interaction for Mobile Platform Virtual Reality Contents. *Symmetry* **2017**, *9*, 22. [[CrossRef](#)]
6. Metcalf, C.D.; Notley, S.V.; Chappell, P.H.; Burrige, J.H.; Yule, V.T. Validation and Application of a Computational Model for Wrist and Hand Movements Using Surface Markers. *IEEE Trans. Biomed. Eng.* **2008**, *55*, 1199–1210. [[CrossRef](#)]
7. Zhao, W.; Chai, J.; Xu, Y.Q. Combining Marker-based Mocap and RGB-D Camera for Acquiring High-fidelity Hand Motion Data. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Lausanne, Switzerland, 29–31 July 2012; Eurographics Association: Aire-la-Ville, Switzerland, 2012; pp. 33–42.
8. Carvalheiro, C.; Nóbrega, R.; da Silva, H.; Rodrigues, R. User Redirection and Direct Haptics in Virtual Environments. In Proceedings of the 2016 ACM on Multimedia Conference, Amsterdam, The Netherlands, 15–19 October 2016; ACM: New York, NY, USA, 2016; pp. 1146–1155.
9. Remelli, E.; Tkach, A.; Tagliasacchi, A.; Pauly, M. Low-Dimensionality Calibration through Local Anisotropic Scaling for Robust Hand Model Personalization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 2554–2562.

10. Joo, H.; Simon, T.; Sheikh, Y. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; IEEE Computer Society: Washington, DC, USA, 2018; pp. 8320–8329.
11. Dong, Z.-C.; Fu, X.-M.; Zhang, C.; Wu, K.; Liu, L. Smooth Assembled Mappings for Large-scale Real Walking. *ACM Trans. Graph.* **2017**, *36*, 1–13. [[CrossRef](#)]
12. Lee, J.; Jeong, K.; Kim, J. MAVe: Maze-based immersive virtual environment for new presence and experience. *Comput. Anim. Virtual Worlds* **2017**, *28*, e1756. [[CrossRef](#)]
13. Marwecki, S.; Brehm, M.; Wagner, L.; Cheng, L.-P.; Mueller, F.F.; Baudisch, P. VirtualSpace—Overloading Physical Space with Multiple Virtual Reality Users. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), Montreal, QC, Canada, 21–26 April 2018; ACM: New York, NY, USA, 2018; pp. 241:1–241:10.
14. Leonardis, D.; Solazzi, M.; Bortone, I.; Frisoli, A. A 3-RSR Haptic Wearable Device for Rendering Fingertip Contact Forces. *IEEE Trans. Haptics* **2017**, *10*, 305–316. [[CrossRef](#)] [[PubMed](#)]
15. Prattichizzo, D.; Chinello, F.; Pacchierotti, C.; Malvezzi, M. Towards Wearability in Fingertip Haptics: A 3-DoF Wearable Device for Cutaneous Force Feedback. *IEEE Trans. Haptics* **2013**, *6*, 506–516. [[CrossRef](#)] [[PubMed](#)]
16. Kim, M.; Jeon, C.; Kim, J. A Study on Immersion and Presence of a Portable Hand Haptic System for Immersive Virtual Reality. *Sensors* **2017**, *17*, 1141.
17. Slater, M.; Usoh, M. Simulating peripheral vision in immersive virtual environments. *Comput. Graph.* **1993**, *17*, 643–653. [[CrossRef](#)]
18. Slater, M.; Usoh, M.; Steed, A. Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.* **1995**, *2*, 201–219. [[CrossRef](#)]
19. Vinayagamoorthy, V.; Garau, M.; Steed, A.; Slater, M. An Eye Gaze Model for Dyadic Interaction in an Immersive Virtual Environment: Practice and Experience. *Comput. Graph. Forum* **2004**, *23*, 1–11. [[CrossRef](#)]
20. Slater, M.; Sanchez-Vives, M.V. Transcending the Self in Immersive Virtual Reality. *Computer* **2014**, *47*, 24–30. [[CrossRef](#)]
21. Roussou, M.; Slater, M. Comparison of the Effect of Interactive versus Passive Virtual Reality Learning Activities in Evoking and Sustaining Conceptual Change. *IEEE Trans. Emerg. Top. Comput.* **2017**. [[CrossRef](#)]
22. Lee, J.; Kim, M.; Kim, J. A Study on Immersion and VR Sickness in Walking Interaction for Immersive Virtual Reality Applications. *Symmetry* **2017**, *9*, 78. [[CrossRef](#)]
23. Visell, Y.; Cooperstock, J.R.; Giordano, B.L.; Franinovic, K.; Law, A.; Mcadams, S.; Jathal, K.; Fontana, F. A Vibrotactile Device for Display of Virtual Ground Materials in Walking. In Proceedings of the 6th International Conference on Haptics: Perception, Devices and Scenarios (EuroHaptics '08), Madrid, Spain, 11–13 June 2008; Springer-Verlag: Berlin/Heidelberg, Germany, 2018; pp. 420–426.
24. Nordahl, R.; Berrezag, A.; Dimitrov, S.; Turchet, L.; Hayward, V.; Serafin, S. Preliminary Experiment Combining Virtual Reality Haptic Shoes and Audio Synthesis. In Proceedings of the 2010 International Conference on Haptics—Generating and Perceiving Tangible Sensations: Part II, Amsterdam, The Netherlands, 8–10 July 2010; Springer-Verlag: Berlin/Heidelberg, Germany, 2010; pp. 123–129.
25. Achibet, M.; Marchal, M.; Argelaguet, F.; Lécuyer, A. The Virtual Mitten: A novel interaction paradigm for visuo-haptic manipulation of objects using grip force. In Proceedings of the 2014 IEEE Symposium on 3D User Interfaces (3DUI), Minneapolis, MN, USA, 29–30 March 2014; IEEE Computer Society: Washington, DC, USA, 2014; pp. 59–66.
26. Achibet, M.; Gouis, B.L.; Marchal, M.; Léziart, P.; Argelaguet, F.; Girard, A.; Lécuyer, A.; Kajimoto, H. FlexiFingers: Multi-finger interaction in VR combining passive haptics and pseudo-haptics. In Proceedings of the 2017 IEEE Symposium on 3D User Interfaces (3DUI), Los Angeles, CA, USA, 18–19 March 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 103–106.
27. Parish, Y.I.H.; Müller, P. Procedural modeling of cities. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01), Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 301–308.
28. Wonka, P.; Wimmer, M.; Sillion, F.; Ribarsky, W. Instant architecture. *ACM Trans. Graph.* **2003**, *2*, 669–677. [[CrossRef](#)]
29. Xiao, J.; Fang, T.; Tan, P.; Zhao, P.; Ofek, E.; Quan, L. Image-based façade modeling. *ACM Trans. Graph.* **2008**, *27*, 1–10. [[CrossRef](#)]

30. Dylla, K.; Frischer, B.; Mueller, P.; Ulmer, A.; Haegler, S. Rome Reborn 2.0: A Case Study of Virtual City Reconstruction Using Procedural Modeling Techniques. In Proceedings of the 37th International Conference (Computer Applications and Quantitative Methods in Archaeology (CAA)), Williamsburg, VA, USA, 22–26 March 2010; Archaeopress: Oxford, UK, 2010; pp. 62–66.
31. Kim, J.; Cho, H. Efficient modeling of numerous trees by introducing growth volume for real-time virtual ecosystems. *Comput. Animat. Virtual Worlds* **2012**, *23*, 155–165. [[CrossRef](#)]
32. Kim, J.; Kim, D.; Cho, H. Procedural modeling of trees based on convolution sums of divisor functions for real-time virtual ecosystems. *Comput. Animat. Virtual Worlds* **2013**, *24*, 237–246. [[CrossRef](#)]
33. Jeong, K.; Kim, J. Event-Centered Maze Generation Method for Mobile Virtual Reality Applications. *Symmetry* **2016**, *8*, 120. [[CrossRef](#)]
34. Jeong, K.; Lee, J.; Kim, J. A Study on New Virtual Reality System in Maze Terrain. *Int. J. Hum. Comput. Interact.* **2018**, *34*, 129–145. [[CrossRef](#)]
35. Prim, R.C. Shortest connection networks and some generalizations. *Bell Syst. Tech. J.* **1957**, *36*, 1389–1401. [[CrossRef](#)]
36. Witmer, B.G.; Jerome, C.J.; Singer, M.J. The Factor Structure of the Presence Questionnaire. *Presence Teleoper. Virtual Environ.* **2005**, *14*, 298–312. [[CrossRef](#)]



© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).