

Real-Time Cloud Modeling and Rendering Approach Based on L-system for Flight Simulation

SeokYoon Kang¹, Kyoung Choon Park² and Ki-Il Kim^{1*}

¹*Department of Informatics, Gyeongsang National University, Engineering Research Institute, Jinju, Korea, 660-701*

²*Aero Master Corporation, Sacheon, Korea*
kikim@gnu.ac.kr

Abstract

Modeling method to render cloud on the simulation is divided into two schemes, static and dynamic, depending on the accessibility of the cloud. Two schemes show the complete dissimilar properties in the aspects of realism and real-time processing. Thus, in this paper, we propose a new scheme that merges two modeling methods of the cloud for flight simulation by using the L-system. Since the proposed scheme considers both realism and rendering speed, it is possible to be implemented the formation and growth of the cloud in real-time manner. Finally, the proposed method is evaluated by the experimental results of real implementation in the flight simulation software based on OpenSceneGraph(OSG).

Keywords: *Clouds modeling, Real-time rendering, L-system, Simulation software*

1. Introduction

Development of simulation software takes different modeling and rendering methods depending according to its own purpose. A cloud is one of the very important elements in the implementation of the external scenes. Specially, flight simulation software requires realistic display, for example, the movement and formation of the clouds, the effect of passing through the clouds. Also, it should maintain acceptable high rendering speed.

Generally, there are two methods for the cloud modeling on the simulation; static and dynamic modeling. While static modeling method try to map the texture of the cloud patterns that already created a cloud model object using modeling tools, dynamic modeling reflects the simulation environment in order to define and render the operation of the physical phenomenon of clouds. Static modeling does not greatly affect the rendering speed because it does not require many operations, except for necessary operations for modeling and texture mapping of the cloud. However, it is less realistic because the effect of cloud is smaller as well as it is very hard to change shape. On the other hand, dynamic modeling has slow rendering speed because of the operation for the physical movement of the particles constituting the cloud. But, it has the advantage of enabling a visual representation of the realistic clouds.

Based on this analysis, existing studies need to be modified in a way to maximize their advantages and compensate the disadvantages by selecting each modeling adaptively. In addition, since the realism and the rendering speed are inversely proportional to each other, it is important to take the advantages of both modeling methods at the same time [1-2].

To achieve this goal, in this paper, we propose a method to take adequate model for the cloud with the L-system which is used mainly in the existing plant growth simulation. The proposed method is to modify the production rule and parameters of L-system. Even though they were used in the existing plant growth, we adjust them according to the form of clouds. The method enables to simulate the growth of the cloud on the real-time

simulation. Also, it can easily create a variety of cloud by changing parameters and the production rule. The method used to model the cloud determines the shape of the cloud by the production rule in a similar way such as static modeling method. And then, cloud effect is implemented by adding the particles, depending on the type of cloud. This is to enrich realism to form a semitransparent cloud model.

In this way, the proposed scheme solves the problem of the existing cloud modeling methods by utilizing extensibility and low complexity of the L-system. Finally, by using the flight simulation software with the OSG, we implement the proposed scheme and compare its performance in several ways. Actual implementation results demonstrate that a variety of cloud model is easily created in low complexity.

The rest of this paper is organized as follows. First, we explain the research background and define the challenges. In the section 2, related work is briefly described. The new modeling method and its implementation are presented in the section 3 and 4, respectively. And then, we make a conclusion in the section 5.

2. Related Work

In this section, we introduce about OSG and L-system csharp library which is related to this work. In addition, we present the research directions through analysis in the point of static and dynamic modeling.

2.1. OpenSceneGraph and L-system Library

OSG is the real-time open source graphics middleware application so application developers use a variety of fields ranging from visible simulation, education, game, medical, scientific to virtual and augmented realities. The OSG supports not only small devices such as mobile, tablet PC but also cross-platform that can run OpenGL ES and OpenGL between a laptop and desktop computer. The OSG is written by using the Standard C++ Standard Template Library (STL) and the scene graph processing method for expressing a high performance and three-dimensional world [3].

It also provides various examples for free to reference and expansion in the Internet. The biggest features of the OSG is an open source. That is, there is no restriction with the use of developed simulation software since it was built based on the open source. So the OSG is expected to be developed in the future. The current performance may be lower than the existing closed approach, but the performance can be improved through continuous updating. Also, standardized software platform developing is possible because it is an open source-based [4-5].

The L-system csharp library provides a source of each of 11 kinds of algorithms and viewer written in C# so the general user can easily test various algorithms of different L-system. Each algorithm has the test data. Test data has been created to model a variety of plants because the production rule and the parameters of the L-system are defined. Edit function of the Viewer can be modified and newly defined for each test data. In addition, the newly defined modeling results can be found easily as shown in Figure 1 [6].

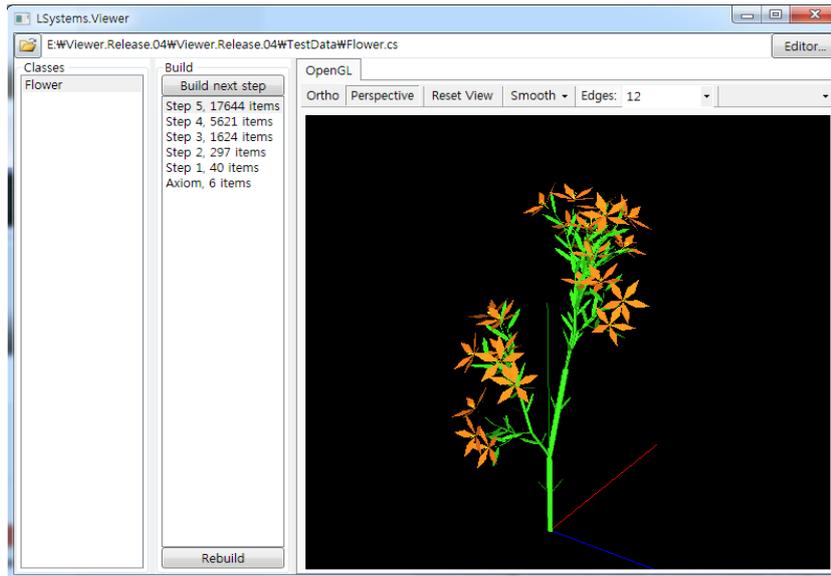


Figure 1. Software Architecture and the Order of Interaction

2.2. L-system

The L-system (Lindenmayer system) is a type of formal grammar as well as description for the structure of various natural objects on the basis of the growth process of the plant, or an algorithm that allows the expression. It is also used for creating a self-similarity shapes and fractal geometry shapes such as Iterated Function System (IFS) in addition to natural objects. The L-system was developed by Aristide Linden Meyer who was theoretical biologist and botanist (Aristid Lindenmayer) of the University of Utrecht, Netherlands in 1968 [7-8]. Since the L-system is the basis of a reflexivity, it is possible to simply describe self-similarity shapes, fractal shapes. In addition, the L-system is simply defined plants and biological structure of natural form. It seems as if the structure is going to grow and complicate by increasing the growing number of recursive calls. The L-system is well used in the creation of artificial life [9]. A simple example using the L-system has the Algae. Algae is the same as the below Figure 2.

Example of L-system(Algae)

- Variables : A, B
- Constants : none
- Axiom : A
- Rules : (A->AB), (B->A)

$n = 0 : A$
 $n = 1 : AB$
 $n = 2 : ABA$
 $n = 3 : ABAAB$
 $n = 4 : ABAABABA$

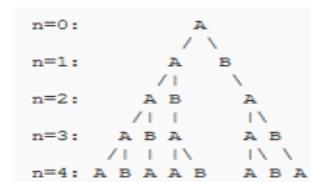


Figure 2. Software Architecture and the Order of Interaction

As shown in Figure 2, algae algorithm is modeled result that is dependent on axiom (initial value A), count of recursive, production rules such as the right graph. Most of the L-system algorithm is similar to the algae, but the difference is the used command, the initial value, the production rule, etc, [10].

2.3. Previous Clouds Modeling Method

This sub-section introduces about the typical techniques and the related research work by static and dynamic modeling methods. There are two evaluation factors for cloud modeling, visual realism and rendering speed of processing. Since two mentioned factors

are regarded to be reciprocal proportion to each other, balanced approach between two parameters is strongly demanded [11-12].

Static and dynamic modeling method have respective advantages and disadvantages in the rendering speed and realism part, because the differences in the rendering technique used in each of the modeling. For example, a typical rendering technique used in the static modeling method is the texture mapping. A texture mapping method is a technique which enables creating an object using a modeling tool, such as 3dmax on an existing form, a natural texture by mapping a model with a small operation when the object is rendered. Previous studies for texture mapping associated with the static model may have an effect partially to increase the realism and more natural texture mapping [13].

However, recent most researches are more concerned in real-time dynamic than static modeling. In the case of dynamic modeling, a good example includes a particle system. Particle system to represent the movements of many particles or simulate the fluid scene is technique of computer graphic for rendering easily than previous techniques. In case of modeling and rendering using particle system, very small and many particles that include physical calculations about movement is represented by shape and movement of object. Due to this property, there are two advantages that can facilitate real-time and realistic rendering, while overload may be occurred by calculation about physical movement when number of particles is increased. Therefore, recent study is focused on solving the overload problems. Based on above analysis, research for static and dynamic modeling is demanded to increase the realism and reduce the load through merge of modeling methods [14-16].

3. Proposed Modeling and Rendering Approaches

For the realistic flight environments, it is essential to provide a variety of clouds effects in the simulator. Also, according to the fact that clouds effect is affected by the volume and wind in its visuality, it is commonly known that there are many common properties between cloud and fog. However, sense of cloud reality is dependent on tons of operation which takes into physical activity cloud of particle and parameter about wind, humidity, temperature and quantity of light account. Particularly, high quality of cloud modeling is not easily achieved due to more complex operation environment. And then the rendering speed and realism in clouds implementation of flight simulation is very important factors.

Since general cloud modeling using L-system is one of static modeling, clouds modeling using L-system has problem of degradation in realistic visuality in near distance. This being so, we propose approaches that to increase realism using particle system, one of the dynamic modeling techniques. For clouds modeling, the main task is to employ cloud modeling using L-system in order to maintain the acceptable rendering speed. More detailed, two approaches are performed to increase the visual realism and rendering speed. One is to introduce production rule and parameter of L-system for clouds modeling. And the other one is to apply the dynamically cloud effect for realistic visualization. Through two schemes, we can maintain the rendering speed as well as enhance reality for cloud at the same time.

3.1. Clouds Modeling Approach Using L-system

The most important task of the clouds modeling using the L-system is to define the production rules and its parameters in accordance with the form of clouds. These two elements used to decide direction of spreading clouds, and the shape, and the density. Therefore, the first thing to do is to determine the shape of clouds to generate. In our previous works [17], we propose how to solve cloud modeling problem by considering sense of reality and operation rendering technique together. Also, rendering time is another parameter to be considered in the proposed scheme. The proposed scheme

introduces L-system to address above requirement. Also, to improve the realism of cloud, particle system is employed according to rules of L-system.

To implement the proposed scheme, first, we take the example of cloud modeling which makes whole shape of cloud through mixed density function implicit function model as the mentioned in equation (1) [18].

$$h_i(r) = \begin{cases} 1 - \frac{4r^6}{9R^6} + \frac{17r^4}{9R^4} - \frac{22r^2}{9R^2}, & (r \leq R), \\ 0, & (r > R) \end{cases} \quad (1)$$

Equation (1) aims to reduce operation time as well as get the similar result as Gaussian distribution. Whole shape of cloud through equation (1) is transformed to L-system. For more detail, we observed that the Equation (1) can be transformed to Turtle interpretation of L-System which is generally used for realistic and various modeling.

In this works, we added type of clouds according to altitude. In general, the clouds are divided into categories of over 10, depending on the shape and height of the appearance. Each cloud is divided into high clouds, middle clouds, low clouds depending on height, and into cumulus, stratus, cirrus depending on the shape. In addition, the shapes or kinds of clouds do not appear as a one type but appear as mixed type. For example, high clouds are formed over height of 6,000m. And, they are divided into cirrus, cirrocumulus, cirrostratus. Middle clouds are clouds formed in height of between less than 6,000m and higher than 2,000m. Altocumulus, altostratus is present within 2,000 ~ 3,000m and there is low clouds within 2,000m above the ground. The other things are can be divided into nimbostratus, stratocumulus, stratus, cumulus, etc, [19]. Position and shape of each cloud is the same as below Figure 2 [20].

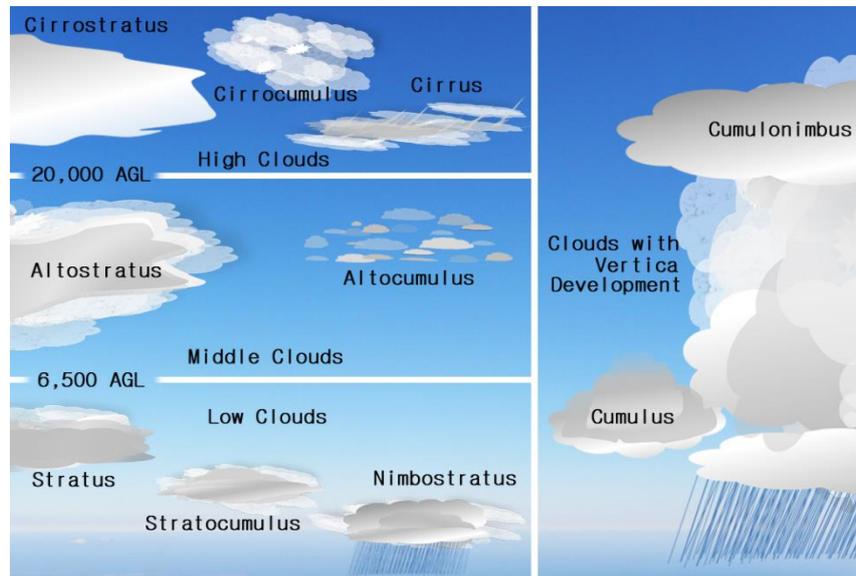


Figure 3. Clouds Types in Accordance with Clouds Layer

We had modeling with choose one of the clouds in each altitude. In addition, we have implemented how to enable the various clouds modeling by entering random parameters value.

The algae algorithm of L-system that is introduced in the above related work section uses a two-dimensional modeling. But, the clouds that are rendered in flight simulation should be a three-dimensional object. To expand it to the three-dimensional modeling, developer should use a turtle interpretation of the L-system. Turtle interpretation has some

command of pitch, roll, yaw direction to the rotation and movable on the three dimensions. Turtle interpretation command is described and defined in below Table 1.

Table 1. Symbols of Command in Turtle Interpretation

Command	Action
F(d)	Move forward a step of length d.
f(d)	Move forward a step of length d without drawing a line.
-(δ)	Turn right by angle δ around the Y axis.
&(δ)	Pitch down by angle δ around the X axis
^(δ)	Pitch up by angle δ around the X axis.
\(δ)	Roll left by angle δ around the Z axis.
/(δ)	Roll right by angle δ around the Z axis.
[Push the current state of the turtle to stack
]	Pop the current state of the turtle from stack
@S(id)	Drawing the surface with the specified id.

In Table 1, Turtle interpretation command of L-System is used for main cloud frame which is for production and cloud shape configuration. Production rules of L-system are defined using the turtle interpretation command such as the above Table 1. We try many time to apply production rules and parameter to the csharp library. As a result, we are able to make each of production rules in accordance with cloud layers.

Table 2. Example of Production Rules and Parameters

Variables	Changed rule
A, a	////[fL&ffLf!AL]////[&ffL!fAfS]////[&ffL!fAL].
F, f	//////.
S, s	fL.
L, l	Create sphere.
Axiom	A, L
LineThickness	The initial size of the sphere, getting smaller..
Distance	Distance of move forward command.
Angle	Angle of direction command.
Depth	Number of recursive call.

We selected the one of the cloud types each layer, and then we were modeling using production rules for each type of cloud. And, we selected the one of the cloud types each layer, and then we did modeling using production rules for each type of clouds. As shown in the Table 2, production rules and parameters are example of one of clouds types. Cloud shape is decided by invoked number from each of defined productions. By invoked number, frame and circle are built.

3.2. Rendering of Clouds Model Approach

In clouds modeling using L-system approach, we propose a method to model a variety of shapes for clouds. However, the proposed modeling method has problem of low realism problem in common with static modeling method. In addition, it is very hard to place various shapes of rendered clouds. To solve this problem, we propose algorithm to increase realism of the clouds model as well as to place the cloud dynamically. A simple flow chart of the proposed algorithm is shown in below Figure 4.

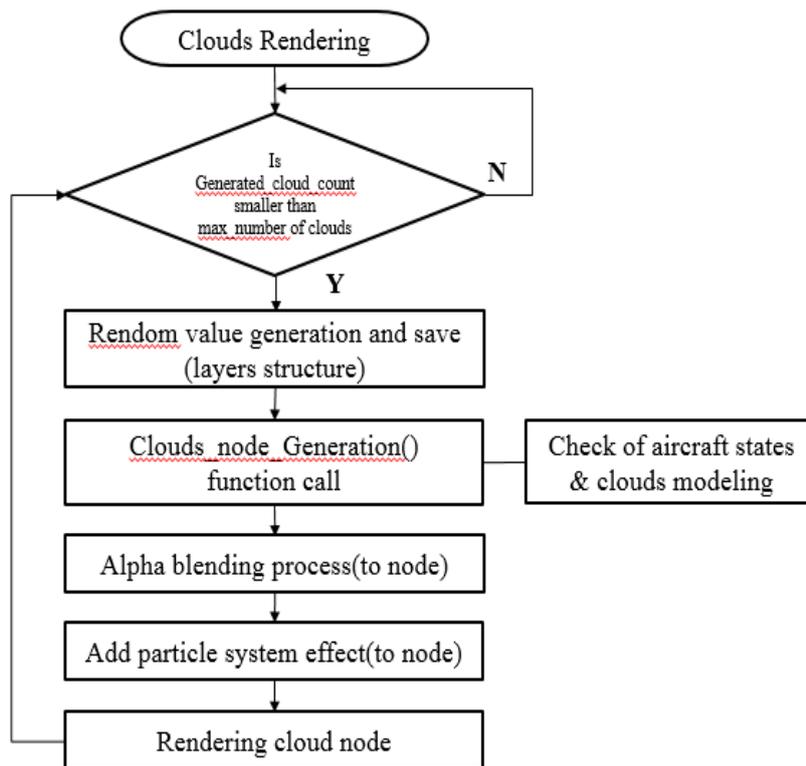


Figure 4. Flow Chart of Clouds Rendering Method

As shown in Figure 4, we generate clouds nodes using the clouds generating function and modeling of clouds object using parameter and production rules of L-system according to each the clouds nodes information. However, the actual modeling results decrease realism because it is part of object of clouds shape consisting of large and small white sphere. For this reason, the clouds object is added in the clouds node as a child node. And then, two techniques in order to increase the realistic effect is applied to clouds by proposed algorithm. The first technique called alpha blending technique, allow cloud objects model in clouds nodes to show the semi-transparent effect. Then, in order to apply effect such as real clouds, we use the second method around cloud objects. The second technique uses a particle system, one of the dynamic modeling techniques. While existing particle system techniques include movement operation of particle, the particle system technique used by proposed algorithm is designed to remove the movement operation as well as fix the particle around cloud object. In this work, since loss of realistic effect is maintained lower than existing particle system that contains movement operation. Also, in case of rendering speed, the proposed scheme is much better than existing particle system.

In order to implement dynamic placement of a variety of clouds in flight simulation, we design function that generates cloud nodes using states values of aircraft before the cloud object. This function is called `Clouds_node_generation ()`. The `Clouds_node_Generating` function is firstly executed in proposed algorithm flow chart of Figure 4 by receiving information of cloud objects. Below Figure 5 presents pseudocode of `Clouds_node_Generation` function.

Clouds_node_Generation

Require : Layer
 Require : aircraft_state

```

1: createclouds_node
2: staticvariablenode_countInitializationto0
3: for i = 0 to node_count do
4:     if Layer[i].node_number = 0 then
5:         Layer[i].node_number = i
6:         node_number = Layer[i].node_number
7:         node_count ++
8:     else
9:         continue
10:    endif
11: endfor
12: function set_structure (Layer, aircraft_state, node_number)
13: clouds_node ← addChild(Clouds_modeling(Layer, mode_number))
14: clouds_node ← setPosition(Layer.Latitude, Layer.Longitude, Layer.Altitude))
15: return clouds_node
    
```

set_structure

Require : Layer
 Require : aircraft_state
 Require : node_number

```

1: if aircraft_state = withinHighLayerrange then
2:     Layer[node_number].ID = 1
3: if aircraft_state = withinMiddleLayerrange then
4:     Layer[node_number].ID = 2
5: if aircraft_state = withinLowLayerrange then
6:     Layer[node_number].ID = 3
7: endif
8: Layer.distance = random number generation(from 0 to 40)
9: Layer.azimuth = random number generation(from 0 to 359)
10: position_set(Layer, aircraft_state, node_number)
11: function set_Parameter(Layer, node_number)
    
```

Clouds_modeling

Require : Layer
 Require : node_number

```

1: create model node
2: if Layer[node_number].ID = 1 then
3:     production_rule = High clouds rule
4: if Layer[node_number].ID = 2 then
5:     production_rule = Middle clouds rule
6: if Layer[node_number].ID = 3 then
7:     production_rule = Low clouds rule
8: endif
9: L – system clouds modeling using production_rule
10: return model
    
```

Figure 5. Pseudocode of Cloud_node_generation function

As the flight simulation is going on, the function receives state information of aircraft in real-time, and records the values on structure of high layer, middle layer, low layer in accordance altitude value of aircraft. Each structure includes the variable information defined in the following Table 3.

Table 3. Variables Information of Layers Structure

Variable	Information
ID	Identify number of layers. High : 1, Middle : 2, Low : 3.
node_number	Identify node.

parameter	Array for parameter values of clouds modeling.
distance	Save the random distance value for clouds generation range.
azimuth	Save the random azimuth value for clouds generation position.
Latitude_pos	Latitude Coordinates for placement of generated clouds model
Longitude_pos	Longitude Coordinates for placement of generated clouds model
Altitude_pos	Altitude Coordinates for placement of generated clouds model
Distance_range	Max range of random value for distance variable. Max : 40 (Unit : Km)
Azimuth_range	Range of random value for azimuth variable. (0~359)
Parameter_range	Array with max value range of the each parameters

As Shown in Table 3, variables are determined by random values that are extracted within set range when generating the nodes. Latitude and longitude coordinates of the cloud is generated within range extracted by using the following equation (2) and (3).

$$\text{Distance} = \sin\left(x1 \times \frac{\pi}{180.0}\right) \times \sin\left(x2 \times \frac{\pi}{180}\right) + \cos\left(x1 \times \frac{\pi}{180}\right) \times \cos\left(\theta \times \frac{\pi}{180}\right)$$

$$\text{Distance} = a \cos(\text{Distance}) \quad (2)$$

$$\pi = 3.1415, \quad \theta = y2 - y1$$

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = (x \sin \theta - y \sin \theta, x \sin \theta + y \cos \theta) \quad (3)$$

Equation (2) is to compute the distance between two coordinates. Since we already know distance and coordinates of x1 and y1 and y2, coordinates of x2 can be computed by inverse equation 2. This process extracts coordinates by random distances from coordinates of aircraft. And, equation (3), rotation matrix equation, extracts rotated coordinates by a certain azimuth on the basis of aircraft coordinates. The procedure to extract the latitude and longitude of Position variables for the deployment of cloud is as follows: First, we extract Temp_X and Temp_Y on coordinates of random distance through the process of inverse equation (2) by latitude and longitude coordinates of the aircraft and the value of the variable range. And then we can extract final coordinates for placement of clouds by substituting the coordinates values (Temp_X, Temp_Y) on coordinates of random distance and random azimuth value in equation (3). Altituded position variable is recorded to random altitude values within the altitude range of cloud Layer by the variable of ID number. Also range of parameter variables varies slightly depending on the identity variable. The range of parameter variables is set to range that can produce other shapes of clouds while maintaining the basic type of the cloud according to the cloud layers.

4. Experiment Results

In order to implement and test our proposed approach, we use our flight simulation software. In addition, we employ csharp library in order to test and make the production rules of proposed clouds modeling more easily. Result of clouds modeling using csharp in accordance with production rules of Figure 1 is shown in Figure 6.

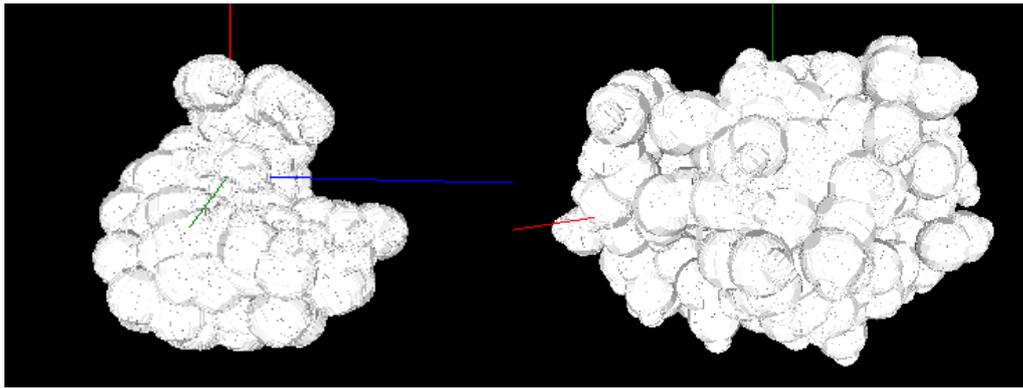


Figure 6. Clouds Modeling Results in Accordance with Production Rules

As shown in Figure 6, we create a production rule of cloud that is modeled on each layer using the csharp library. To increase the realistic effect for the model, we apply the proposed methods in our flight simulation software. Because available L-system libraries in OSG are not available now, we add a new L-system library that is based on turtle interpretation employed in our flight simulation software. By applying the proposed approach on OSG, we render clouds object with increased realism. The results are shown in the following Figure 7

Left side of Figure 7 shows the effect of applied method to increase the realism. On the right is the final result by applying rendering algorithms to the result on the left. The below Table 4 represents results of performance evaluation proposed approach and previous methods by measuring Frame Per Second (FPS).

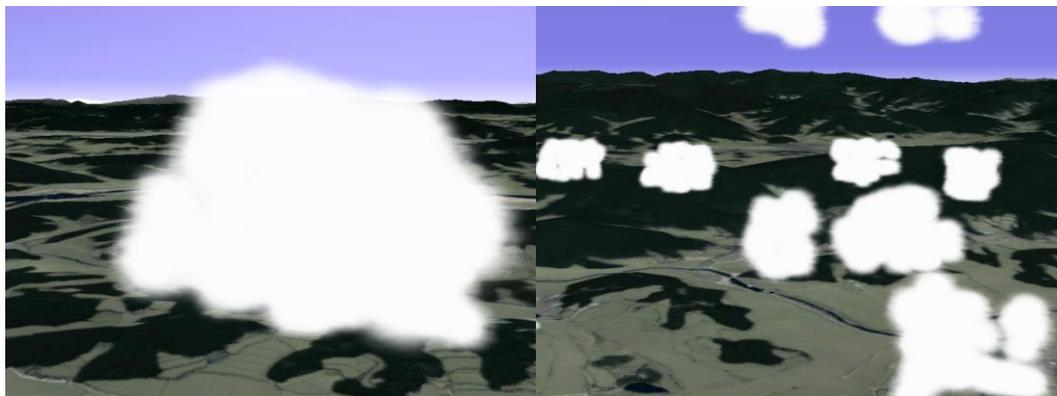


Figure 7. Applied Approach Results of on Flight Simulation Software Using OSG

Table 4. Average Values of Frame Rate in Accordance with Modeling Method

Distance\ Method	Dynamic Modeling	Proposed Approach	Static Modeling
Far	50.46 Hz	60.00 Hz	60.00 Hz
Near	15.85Hz	59.43 Hz	60.00 Hz
In clouds	5.21 Hz	52.74 Hz	60.00 Hz

Each test results have been measured ten times

Through the results shown in above both Figure 7 and Table 4, we make rough conclusion that our approach have better performance in a realistic and rendering speed than a conventional method. Reality is better than static modeling, but performance decrease is minimal among them. Conversely, realism is better than dynamic modeling, but the performance is improved highly. Also, frame rate of dynamic modeling is reduced in all cases. When the range of dynamic modeling particle is widened, graphical computation increases. Through these results, we confirmed that the realism and performance of the proposed approach are improved.

5. Conclusion

In this paper, we proposed new cloud modeling technique by making use of outstanding features of the L-system as well as considering simplicity and scalability. For the improved sense of reality, even though we employed the particle system, acceptable overhead of particle system is maintained by L-system. In additional, we applied clouds generation function in accordance with positions of aircraft for rendering in real-time way. Finally, we confirmed the possibility of applicability through the experimental results on OSG.

Related to this work, we will continue the research and implementation for different L-system methods to apply a cloud model and derive the general transform principle for easy conversion. Furthermore, we will study how to enhance the realism and rendering performance.

Acknowledgements

This research was supported by SW Master's course of hiring contract program grant funded by the Ministry of Science, ICT and Future Planning and Production Technology Commercialization Support program by the Korea Industrial Complex Corporation.

References

- [1] Upchurch, E. M, Semwal, S. K, "Dynamic cloud simulation using cellular automata and texture splatting", Proceedings of Computer Simulation Conference, (2010) July 11; San Diego, USA
- [2] H. Cui, M. Qi and D. Li, "3D cloud modeling based on fractal particle method", Proceedings of Electrical and Control Engineering (ICECE). (2011) September 16-18; Yichang, China.
- [3] About features, OpenSceneGraph, <http://www.openscenegraph.org/index.php/about/features>
- [4] R. Wang and X. Qian, "OpenSceneGraph 3.0 Beginner's Guide", PACKT, Birmingham (2010).
- [5] R. Wang and X. Qian, "OpenSceneGraph 3.0 Beginner's Guide", PACKT, Birmingham (2010).
- [6] Example of L-system, Lsystem-csharp-lib, <https://code.google.com/p/lsystems-csharp-lib/>
- [7] A. Lindenmayer, "Mathematical models for cellular interaction in development", Theoretical biology, vol. 3, no. 18, (1968).
- [8] D. Fang and X. Li-Feng, "An Application of L-System and IFS in 3D Fractal Simulation", WSEAS Transactions on systems, vol. 1, no. 7, (2008).
- [9] R. Sun, J. Jia and M. Jaeger, "Intelligent tree modeling based on L-system", Proceedings of Computer-Aided Industrial Design & Conceptual Design, (2009) November 26-29; Wenzhou, China.
- [10] L-system, Wikipedia, <https://en.wikipedia.org/wiki/L-system>
- [11] X. Hu, S. Bo, Z. Huiqin, X. Bing, W. Hao and H. Ge, "Visual simulation system for flight simulation based on OSG", Proceedings of Audio Language and Image Processing (ICALIP), (2010) November 23-25; Shanghai, China.
- [12] I. Strachan, "Jane's simulation and training systems Janes Information Group", (2009).
- [13] M. J. Harris, "Real-Time Cloud Simulation and Rendering", The University of North Carolina at Chapel Hill (2003).
- [14] J. Xu, C. Yang, J. Zhao and L. Wu, "Fast Modeling of Realistic Clouds", (ICECE), Proceedings of Computer Network and Multimedia Technology (CNMT), (2009) January 18-20; Wuhan, China
- [15] G. Sellers, R. S. Wright, N. Haemel, "OpenGL SuperBible: Comprehensive Tutorial and Reference", Addison-Wesley, United States (2013).
- [16] H. Wang and F. Meng, "Real-time Simulation of Dynamic Clouds Based On Cellular Automata", Hybrid Information Technology, vol. 5, no. 6, (2013).

- [17] S. Y. Kang and K.-I. Kim, "Three Dimensional Cloud Modeling Approach Based on L-System", Proceedings of Computer & Information and Application (2015) May 22-25; Yeosu, Korea.
- [18] J. Y. Do, N. H. Baek, C. W. Lee and K. W. Ryu, "Modeling and Rendering of Clouds for Real-time Flight Simulation", Journal of information processing society, vol. 5, no. 16, (2009).
- [19] Houze Jr, and A. Robert, "Cloud dynamics", Academic press, British (2014)
- [20] Cloud Types, oldWeather, Intel Corporation, <http://forum.oldweather.org/index.php?topic=3003.0>

Authors



Seok Yoon Kang, he received the B.S. and M.S. degrees in informatics from Gyeongsang National University. He is currently working toward Ph.D. degree at Gyeongsang National University. His research interests include Flight Simulation Software.



Kyoung Choon Park, he received the M.S. degree in computer science from Gyeongsang National University. He is currently working at Aero Master Corporation. His research interests include avionics software, software engineering and signal processing.



Ki-II Kim, received the M.S. and Ph.D. degrees in computer science from the ChungNam National University, Daejeon, Korea, in 2002 and 2005, respectively. He is currently with the Department of Informatics at Gyeongsang National University. His research interests include routing for MANET, QoS in wireless network, multicast, sensor networks and avionics software.