Journal of Image and Graphics, Vol. 9, No. 3, September 2021

Visual Restoration of Ocean Wave Effects Produced by Houdini in the Engine

Jiani Zhou and Tae Soo Yun
Department of Visual Contents, Dongseo University, Busan, South Korea
Email: {moses472530, yntaesoo}@gmail.com

***Abstract*—In this paper, we propose importing ocean wave effects produced by Houdini into the game engine through plug-in form and displaying it through real-time rendering, and the optimal value setting of the plug-in in terms of animation completeness and detail reduction degree is verified by the way of example production. We use ocean wave effects produced by Houdini in an Unreal Engine as the research theme. Film and games are two important players in the entertainment industry. The game focuses on the interaction with a player, whereas the focus of the film is the expression of the picture. This also makes the game and the film to have very different performance characteristics in the early stage of development. It is worth noting that the boundaries between film and video have become increasingly blurred in recent years. At the cultural level, there are cases of integration of film and video in IP cooperation, and narrative techniques also have to improve efficiency. The significance of the optimal value is that this value can restore the visual effect of Houdini to the maximum effect. The introduction of the game engine provides a more convenient and real-time display and a wider application scenario for the wave effect produced by Houdini. The form of plug-in realises the binding of the film special effects software Houdini and the game engine. It also confirms the general trend of mutual integration of games and movies from the technical point of view.**

***Index Terms*—ocean wave effect, Houdini, engine, VFX**

I. Introduction

In this era of cross-industry cooperation aimed at achieving a win-win situation, film and television works are not only the film production company's full power; the rise of the game company has become a fresh blood in this field and cannot be ignored. At the level of special effects production, the main difference between games and film is the difference in picture quality. Film effects are built on powerful time and price costs, and the price/time ratio used for movie rendering is hundreds of times that of a single PC (game) [1].

After constant evolution, today's game engine has evolved into a complex system consisting of multiple subsystems, covering almost every important part of the development process, from physical systems to modelling, animation, lighting and particle specialties to file management from crash detection, network characteristics and professional editing and plug-ins. Machinima refers to a computer graphic video produced based on a 3D game engine, a combination of computer machines and cinema [2]. Compared to current CG and 3D animation production plants, it has the advantage of being able to produce them in less cost and faster time and is being used for various purposes such as pre-visualisation of films, Cut Scene in games and User Generated Contents (UGC) [3].

The ocean wave effect has played a crucial role in many movies. Its randomness and complexity take a long time and much hardware equipment to simulate a sufficient realistic ocean wave effect. For 3D software Houdini, as an example, the conventional ocean wave effect is divided into large and small collision-free zones; real marine specialties must be simulated through the dynamic system; and several simulations are necessary if the camera undergoes a series of wave-induced wave testing, collision simulation, lighting and material, renderings and so on. This process is very time consuming.

The production of ocean CG special effects requires a visual analysis of the actual sea, and it is necessary to understand that objects' drift by the action of waves is the action of wind and sea currents. The melting waves often involve bubbles that remain in many shocks and are sharp, fine white blisters. A level sea surface has relatively small waves and no water buffalo, and only a small amount of foam is produced on it. Fig. 1 indicates that the special effects of achieving the sea are generally divided into four parts: volume, surface, foam and mist [4].

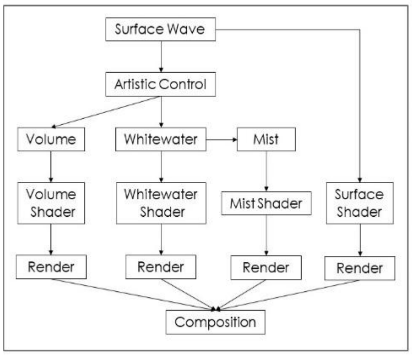

Figure 1. The schematic diagram of ocean special effects by Houdini.

Among computer graphics, simulations and studies of spongy waves mainly have several methods: Peachey (1986) used geometric methods, i.e. sine curves, bezie curves and so on, to construct waveforms of waves, and

Manuscript received December 21, 2020; revised June 29, 2021.

doi: 10.18178/joig.9.3.109-113

to construct the shape of waves by means of different phases [5]. Fornberg and Merrill (1997) is a hydrodynamic calculation, primarily using the Finite Difference Method to obtain an answer similar to the Navier–Stokes equation and to obtain a wave altitude field using the obtained speed and pressure field to simulate the sleep effect [6]. Yen Lai-bin (2000) used a cloning method based on a wave spectrum, mainly linear overlay and linear filtration, producing waves of different wave classes through different function structures [7]. The method of simulation based on texture is to adopt a bump map, in which the variation of the surface legal line (bump map) results in different degrees of intensity at each point of the surface through the variation of the legal line (bump) under the irradiation of the light; that is, it results in the formation of a wave-like shape. The above methods can also be returned in physical and structural ways. Eun-Ju Yang (2003) used statistical models and imaging techniques through observation of oceanography and used FFT technology to acquire wave patterns and simulate deep ocean [8]. Dongmin Lee (2006) based on the Gerstner model and the spectral model, the ocean is represented by a low-resistance non-uniform grid and a high-resolution uniform grid normal texture generated in the graphics hardware [9]. In the interactive ocean, Li yin (2017) improved the large-scale ocean's interaction effect and real-time performance by using a GPU to construct the ocean surface, introducing the PhysX physics engine according to the real world mechanics to simulate the real floating object motion state, and realised the gesture control ship based on the Kinect somatosensory interaction. In the process of the sea surface, the flow of this whole system has strong theoretical and application value [10].

Min Ju Park (2019), through vertex animation, used the adhesion force model employed in textile research based on real world experiments and considered the effect of wrinkles [11].

In research at the VR level, Jiyoung Kang (2018) drew characteristics of the pre-render VR animation pipeline by comparing them with the production pipeline of the existing 3D animation to produce the VR animation efficiently [12]. At the same time of technical research, some scholars conducted theoretical research. Sung Ryong Hong (2014) sorted out the side effects of VR games and proposed countermeasures [13].

Zhou (2018) divided the sea into four parts: volume, surface, foam (white water) and mist [4]. From a downward angle, the surface of the sea can be simulated with a plane with vertex animation, produced visual effects similar to normal, such as reflections and refraction with later additional materials and lights, and 'Fake' can create a volumeless ocean. However, how to bring the engine has not been realised. Engines can buy existing ocean scenes or materials at stores, but modifications also take a long time and much effort for beginners who do not have the basis of engine blueprints. Thus, if an artist can introduce the ocean wave effect created by Houdini into the engine and verify the optimal camera lens effect through the real-timeisation of the engine, this will not only be widely available in the pose-production phase of the game video but also be output directly from the machine. The special effects zone of the engine must be an important enhancement if it provides a wider display stage and application range to the special effects created by Houdini.

Therefore, the second chapter of this paper first examines the research status of the ocean wave effect and two implementations of special effects in the engine, has an understanding of the overall process, and describes the simulation of the ocean wave effect based on vertex animation. The third chapter is the test phase. After importing the engine through the ocean wave effect completed in Houdini, the polycount test is searched for the optimal value for future research.

II. Implement Engine Effects

The special effects of pyro among Houdini, a 3D special effects programme, are the strengths of the programme, focusing on flames and dance. Fabrication features include dynamic simulation, form creation and rendering. Form creation is mainly controlled by the accuracy, distribution situation and size of the source. The main elements of dynamic simulation control are the speed field of the source, some parameters on the pyro solar node in the Dop (Dynamics OPerations), the accuracy of the Smoke object, and effects of other fields added during the Dop. Fig. 2 illustrates the principle of pyro special effects in Houdini. Thus, the realisation of the pyro special effects among Houdini requires higher conditions for the VFX artist's professional skills and art symbols.

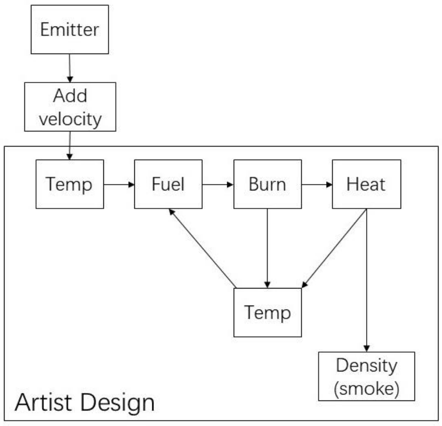

Figure 2. Principle of pyro special effects in Houdini.

In the engine, there are two ways of expressing the form; one is to use the material to express, and the other is to directly give the model, according to special effects to be presented. One of the more interesting features of the flame is that its shape is not fixed, and its movement speed is fast. The expression of the material in the engine is faster and more effective for us to achieve the effect we need. Fig. 3 indicates that the flame effects are presented in a joint state by the particle system, superimposed from left to right. Take the flame below as an example. There are six levels, but the left picture of Fig. 4 indicates that the flame is mainly on the bottom layer, covered with smoke, and the spark is at the top; we can see the simulation of the flame effect on the right.

Figure 3. The joint chart in the engine.

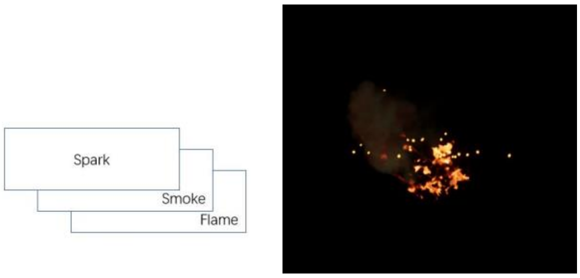

Figure 4. Simplifying the relationship between Mars, smoke, and fire (left); simulating the flame effect (right).

During the Ureal Engine, flame special effects can be realised by using a self-mounted particle system and by a blueprint. Blueprints can be linked to complex visual effects such as variables, functions and events, through links in a form of visual script, but they need support for flame texture. The blueprint enables the effect of the flame to be realised using two methods: texture and sequence plot. The solitary texture method satisfies visual diversification through adjustment of the basic properties of a single texture (rotation and zooming) so that the effect of the flame is not too monotonous. Fig. 5 illustrates two approaches to making a flame map. Flame material realised by sequence typically requires the arrangement of the first figure in the form of nine-to-six-character characters, etc., and a large combined figure will be equipped with an alpha channel. That is, the background of the picture must be black. Among the flame effects produced in recent years, the use of flipbook function nodes is like a strainer, literally turning over a bookcase [14].

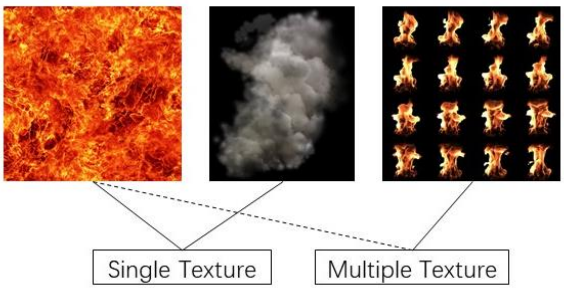

Figure 5. Two approaches to making a flame map.

Fire material can be implemented in two ways, and smoke material can also be obtained. The engine is identified and used by the engine only by a single picture or sequence to represent the flame special effects with Houdini to achieve flame effects in a particle system.

The second method is the vertex animation obtained by recording the motion trajectory of the model points. Storing complex animation data in 2D textures or grid UVs is a great way of reducing animation overhead while maintaining the necessary animation look and feel. However, the Vertex Animation Tools in the Engine every 2D texture affects up to 8,192 vertices [15]. From Houdini, it can directly output an exr file of all vertex positions through the game development tool and record the required point information.

III. Process from Houdini to Engine

A. *Software Equipment Foundation*

When testing this import early, one may find that the details one sees in Houdini have many missing traces after importing of the Engine because of material loss. Based on the computer configuration, even if it is not required to fully inherit the details presented in Houdini, I hope that the big wave will now not be ignored. The version of Houdini is 17, which is the need required game development tool (plug-in), and the version of the Unreal Engine is 4.18.3.

B. *Settings Made in Houdini*

In Houdini, we choose to be able to collide with objects and can create a guided ocean in infinite waters; the type of ocean chosen is Houdini's own, so such tests have certain universality and can be widely used. Sampling is set to 60m length × 60m width, subdivided into 500×500, for the shape of the ocean wave, through five different spectra to ensure that the entire wave has enough details. Fig. 6 indicates that the state after the sea surface changes. Additionally, the number of particles sampled in the sea is 300,000, because we need to export in the form of mesh; for the subsequent adjustment, the sampled sea area is cached, which caching takes one hour and six minutes.

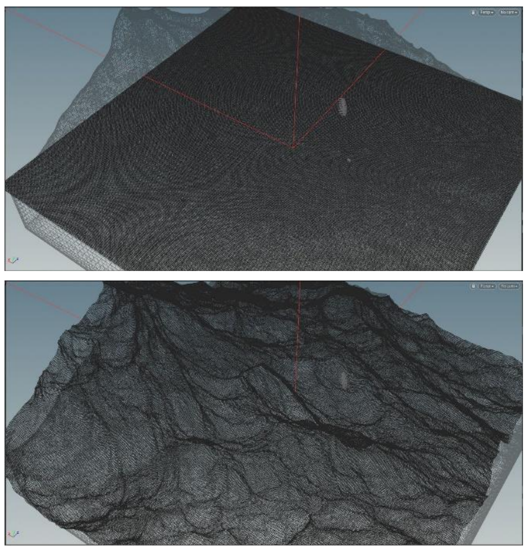

Figure 6. The state after the sea surface changes.

C. Settings in the Engine

The animation of 119 frames produced by Houdini is matched with the original model after importing of the engine. A file that has been exported with a position can directly output an exr file of all vertex positions through the game development tool and record the required point information.

We hope that the details produced in Houdini are more likely to be completely imported into the engine. In Houdini's game development tool, the optimal value is tested by changing the value of polycount. After the polycount was given five values of 8,000, 10,000, 30,000, 60,000 and 100,000, the model was broken in the ocean of 60,000, so a limit of 45,000 was added. Here we visually understand their status through a top view perspective.

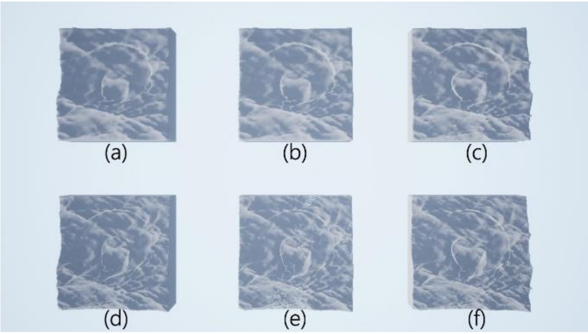

Figure 7. The state of the six test oceans at frame 1.

From Fig. 7 to Fig. 10, four stages have been selected that have changed since the cinematics screenshot. In the figure, a, b, c, d, e, and f represent polycounts of 8,000, 10,000, 30,000, 60,000 and 100,000, respectively. Fig. 7 illustrates the state of the six test seas in the first frame, all of which are in a stable state. It can be seen that the details of (d), (e) and (f) test areas are more than those of (a), (b) and (c) test areas. As illustrated in Fig. 8, at the beginning of the 53rd frame, the (f) test area exhibits a rupture, and the animation is still going on, but the model has no way to maintain it.

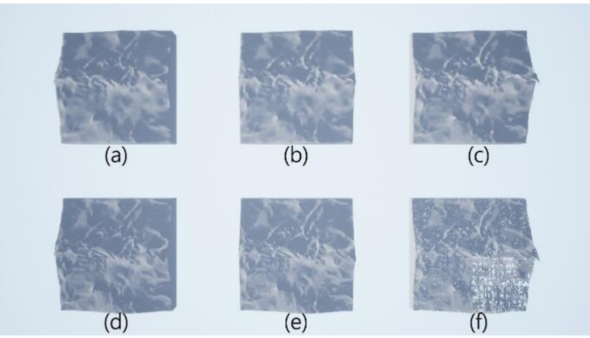

Figure 8. The state of the six test oceans at frame 53.

In the 91st frame, the (e) test area with a polycount value of 60,000 starts to rupture, and at this time, the (f) test area is completely out of shape, with a flicker flickering, as illustrated in Fig. 9. In Fig. 10 of the 116th frame, the (e) test area begins to be out of shape, with a flickering of the snowflake, and the (f) test area at this time has completely disappeared.

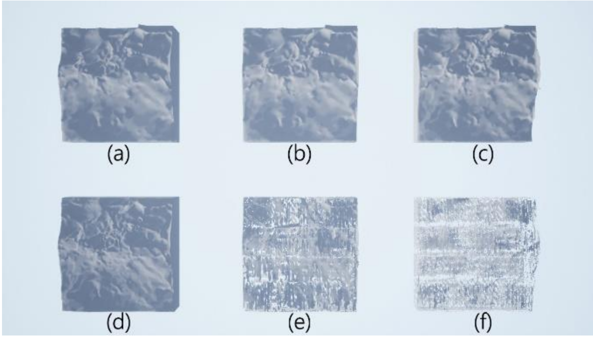

Figure 9. The state of the six test oceans at frame 91.

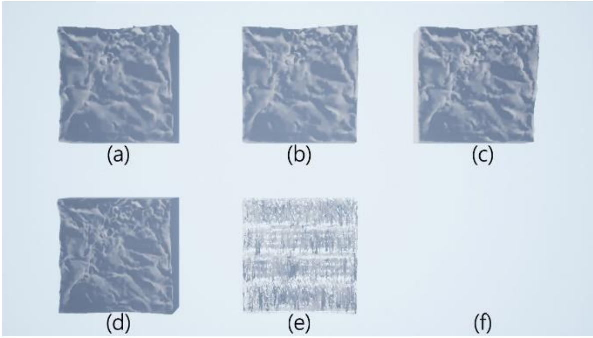

Figure 10. The state of the six test oceans at frame 116.

D. Result

We wanted to ensure that the desired effect is to import the ocean wave effect into the engine in Houdini. Even if one does not need to fully inherit the details, one wants to keep the big waves or ensure that the body is not lost. Table I organises the target polycount of the six test areas, the export time and the state in the engine. The status in the engine in Table I records whether there is continuous animation (A) and whether the model is broken (B). If the effect has a continuation animation and no model is broken, then it is what we finally want.

In these 116 frames, (a), (b), (c) and (d) test areas have a good maintenance in the continuity of the animation and the integrity of the model. Test areas (e) and (f) are broken at different time points; the point of the ocean wave effect made in Houdini is 300,000; the derived position texture is 4K, which is the result after importing of the engine; and the current value indicates that the target number of 45,000 is the best value. The (e) test area was used as a benchmark in subsequent studies.

TABLE I. SIX TEST AREAS' INFORMATION

Name	Target Polycount	Export Time	Status in the Engine
a	8,000	20 min	A(o)B(x)
b	10,000	22 min	A(o)B(x)
c	30,000	23 min	A(o)B(x)
d	45,000	22 min	A(o)B(x)
e	60,000	19 min	A(o)B(o)
f	10,000	21 min	A(o)B(o)

IV. Conclusion and Outlook

Based on the ocean wave effect made by Houdini, this study uses the vertex animation texture node in the game development tool to import the fluid into the engine in mesh form using vertex animation. The goal of the study will be tested on the maximum polycount ceiling of the ocean wave import engine made in Houdini. Although the point of the ocean wave effect made in Houdini is 300,000, the derived position texture is 4K, the result after importing of the engine. The current value indicates that the target number of 45,000 is the best value. In the following research, different scene settings will be introduced.

In Houdini, for the distance to the centre of the ocean surface, the engine uses LOD and set to three stages, which correspond to the scene in the simulated Houdini, compares whether the time and quality are consistent, etc. and then puts forward their own opinions on different situations and different scenes. In the subsequent research, the export time of the optimal value can be used as the time standard and the way Houdini is rendered by the sequence frame.

Compare the time required. Through two methods of comparing the time and the effect of the picture, combined with the concept of the near-middle range, it is proposed that the idea of LOD be used to ensure the wave effect for different application scenarios. Process and summarise the intuitive digital reference standards. This reference standard is beneficial to optimisation of the original wave effect production process.

Conflict of Interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Author Contributions

At the beginning of the paper, Jiani Zhou and Prof. Tae Soo Yun discussed the topic, preparing for the research to be conducted. The professor guided the overall review and research method, and Jiani Zhou started the test under the guidance of the professor and recorded all the data to be used. When the test results were obtained and the text was written, the professor gave much feedback. All authors have approved the final version of the manuscript.

References

[1] Why can't game effects be as real as in movie effects? [Online]. Available: http://www.360doc.com/content/14/0327/19/14106735_364234528.shtml

[2] M. Picard, "Machinima: Video game as an art form," *Author & Canadian Games Study Association*, vol. 6, no. 1, 2006.

[3] Y. J. Park, "A study of visual media scalability by Machinima in MMORPG," *The Korean Society for Computer Game*, vol. 25, no. 2, pp. 7-14, 2014.

[4] J. N. Zhou, "Realization of realistic wave effect in Houdini based on bird's-eye view," in *Proc. International Conference on Cultural Technology*, 2018, pp. 335-338.

[5] D. R. Peachey, "Modeling waves and surf," in *Proc. the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 1986, pp. 65-74.

[6] B. Fornberg and D. Merrill, "Comparison of finite difference- and pseudospectral methods for convective flow over a sphere," *Geophysical Research Letters*, vol. 24, no. 24, pp. 3245-3248, 1997.

[7] L. B. Yan, S. K. Li, and S. S. Zhang, "Real-Time generation of ocean wave surface," *Journal of Computer Aided Design and Computer Graphics*, vol. 12, no. 9, pp. 715-719, 2000.

[8] E. J. Yang, E. J. Kim, and K. W. Ryu, "FFT of deep ocean simulation," *Korean Institute of Information Scientists and Engineers*, 2003, pp. 664-666.

[9] D. M. Lee and S. K. Lee, "Real-Time ocean wave simulation and rendering using GPU," *Korean Institute of Information Scientists and Engineers*, 2006, pp. 91-93.

[10] Y. Li, Y. Tang, F. D. Fan, J. X. Meng, N. An, and J. X. Peng, "Real-Time simulation of large-scale ocean scene in the somatosensory interactive extension," *Journal of Chinese Computer Systems*, vol. 38, no. 3, pp. 590-594, 2017.

[11] M. J. Park and H. W. Byun, "Wet cloth animation with vertex-based adhesion force model," *Journal of the Korea Computer Graphics Society*, vol. 25, no. 1, pp. 1-11, 2019.

[12] J. Y. Kang and S. I. Choi, "A study on pipeline for effective production of VR animation," *Journal of Digital Contents Society*, vol. 19, no. 5, pp. 971-979, 2018.

[13] S. R. Hong, "Countermeasures for side effects of online virtual reality games," *Journal of Digital Contents Society*, vol. 15, no. 3, pp. 405-412, 2014.

[14] UE4 plays picture sequence frame. [Online]. Available: https://gameinstitute.qq.com/community/detail/121660

[15] Vertex animation tool. [Online]. Available: https://docs.unrealengine.com/zh-CN/Engine/Animation/Tools/VertexAnimationTool/index.html

Jiani Zhou was born on 30 July 1994, in Jiangsu, China. She is a doctoral student in visual contents at the Dongseo University of Korea. She received her B.S. in Animation from Zhongnan University of Economics and Law, China. Since 2015, she has been studying in Dongseo University of Korea. She received her M.S. in the Department of Visual Contents from Dongseo University of Korea in 2017. Her current research interests include VFX, image processing, game ethics and visual communication.

Prof. Tae Soo Yun was born in Pohang City, Korea, in 1968. He received his B.S., M.S., and Ph. D. degrees in Computer Engineering from Kyungpook National University, Korea, in 1991, 1993 and 2001, respectively.

He is currently working as professor in the department of digital contents, Dongseo University of Korea from 2001. Dr. Tae Soo is a chief vice president of IACST (International Association for Convergence, Science and Technology) and is in charge of the center for AGRIC (Arcade Game Regional Innovation Center) of Dongseo University. His current research interests include game technology, artificial intelligence, virtual reality, and interactive media.