Optimizations of VR360 Animation Production Process

Wei-Chih Liao(✉), Chun-Tsai Wu, and Szu-Ming Chung

Department of Digital Content Design, Ling Tung University, 1 Ling Tung Road, Taichung 408, Taiwan, ROC

Abstract. The conventional 3D animation production process often takes place through a design stage, including modeling, mapping, animating, and rendering. A frame of movie quality 3D animation can take tens of minutes or even hours to compute and render. In these cases, a large-scale render farm is often utilized to perform rendering and save time and money. Owing to advances in technology, virtual reality (VR) has become a current trend in animation film production. Whether with a 360° panoramic camera recording or 3D technology, many artists and software developers are continually researching more creative forms of expression in this field. Compared with panel-view 3D animation, those in VR animation production must deal with the spherical images of a virtual space in 360°. Details are intensively emphasized. An 8-K or even 4-K rendering resolution provides substantially enhanced viewing quality, but the rendering workload increases substantially. Recently, game engines have developed progressively. Real-time rendering GPU performance, compared with nonreal CPU clusters, has effectively improved the production process, especially regarding instantaneous preview and editable features. Through a creative production of VR360 animation, we optimized the process for a small team. The optimization of the VR360 animation production process comprises seven stages: initial modeling, 3D concept design, and details, texture drawing, rigging, motion capture, motion adjusting, and final integration.

Keywords: VR animation production process · Real-time rendering · Unreal game engine

1 Introduction

Conventionally, a 3D computer-based key-frame animation is based on hand-drawn animation. It remains the main method by which to create 3D animation today. Virtual reality (VR) animation and video is a new, expressive form for media artists to present their creative artworks. The recent prosperity of the gaming industry has led to prolific VR games and game animation products. To take advantage of the real-time rendering capabilities of a game engine, without sacrificing the fidelity of live performance and by saving time and money, in the present study, a team of technology artists examined and improved animation workflow by creating a VR360-based animated short film. This paper presents an optimized VR animation short film production process. First, the

A. Brooks et al. (Eds.): ArtsIT 2020, LNICST 367, pp. 320–330, 2021.
https://doi.org/10.1007/978-3-030-73426-8_19

relevant literature is reviewed. Second, the process of creating a VR360 animation short film is detailed, and many fundamental concepts and procedures are discussed. Finally, the results of an interview with a team leader and three technology artists to discuss technical problems and solutions to optimize the production process are presented.

2 Literature Review

2.1 VR Animation Production Process

Many influential animated films and their characters, such as Pixar's *Toy Story* (1995), *Finding Nemo* (2003), *Up* (2009), *Toy Story 3* (2010), and *Monsters University* (2013); George Lucas and Industrial Light & Magic's *Star Wars* (1999–2005); and *Lord of the Rings*' iconic Gollum character (2001) all took a considerable length of time to render and thousands of computers to complete final production [1]. Many small production studios are wary of investing such an abundant amount of assets and time to create 3D animations. The most popular game engines, such as Unity 5 and Unreal 4.2, can now be used for VR game and animation creations. They possess substantial graphical technologies and real-time rendering capabilities. Unreal game engines can be supported by many animation production tools and are compatible with Maya, a 3D software program. Unreal engines with the capabilities of VR360′s motion capture and real-time photorealistic rendering are ideal tools for integrating 3D animation software and, for the first time, enable animation artists to create short artistic VR animation films at a low cost and within a reasonable time frame.

2.2 Real-Time Rendering

Although real-time rendering is mostly used in interactive gaming, it also allows artists to instantly preview the final results throughout the entire working pipeline. Using a physically based rendering (PBR) shader in an unreal game engine, artists can obtain high-quality rendering results and avoid programming in Unity, which lowers resolution geometry and textures, has less complex rigs, and reduces the requirement for expensive calculations from the rendering engine [2].

2.3 Motion Capture and Live Performance

Motion capturing can produce realistic animation. This process uses cameras to record live performances and map them onto rig characters. Applying such a procedure to drive animation is referred to as performance animation. A computer puppetry system must capture the live performance, accurately interpret and map a performer's motion onto the target digital character, and finally, make the digital character mimic the performance. Such processing can lead to low fidelity [3] and a lack of similarity between the performer and digital character. Even overcoming a seemingly minor obstacle, such as a character disconnectedly interacting with objects in a scene, and the subsequent retargeting that is required is a nontrivial task [4]. Shin et al. [5] proposed an approach that maps input based on an analysis of the root position of the character, joint angles, end effector positions, and the distance of the character to an object in the scene.

3 Design Content

3.1 VR360 Animation Short Film

The Abandoned Deity is a VR360 Animation Short Film. The story brings the audience back to the 1980s to early 1990s. For the first time, the viewers can wear a VR device and watch the old time with the gods' perspectives. We as gods witness humans' fallen religious belief and realize it is human to be abandoned by gods.

The protagonist of the film, Xiaolun, is the son of a famous master of local Buddha idol maker shop. The main scene of this short film occurs in a traditional craftsmanship family making Buddha idols.

3.2 Art Design

The Abandoned Deity was filmed in the Sanmin and Yancheng Districts of Kaohsiung in South Taiwan. The art design was based on the documentary and graphical data of the late 1980s and early 1990s. The characters' appearances and costumes were developed by referencing photographs and works of art from that period. For example, the game room in the work is designed with reference to the 80s style of the movie 'Rebels of the Neon God' (Fig. 1) [6].

Fig. 1. Game room in 1980s, image retrieved from "Rebels of the Neon God"

3.3 Modeling and Materializing

Maya is the most popular 3D animation software program. Conventional 3D animation production uses Maya for modeling, materializing, binding, animating, lighting, and rendering. Its technology is highly developed with advanced processing techniques and controllability. Although it is occasionally insufficient, requiring additional software to accelerate the animation process and improve quality, Maya was adopted here to construct the preliminary modeling for our production. ZBRUSH was used to assist with 3D concept design and detail processing, and Substance Painter was used to creating PBR material texture. Finally, the converted FBX file was imported to Unreal for integration (Fig. 2, 3 and 4).

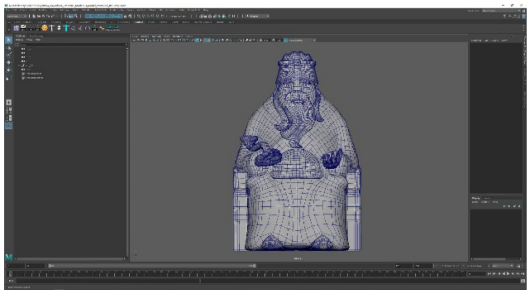

Fig. 2. Maya-topology and UV disassembly

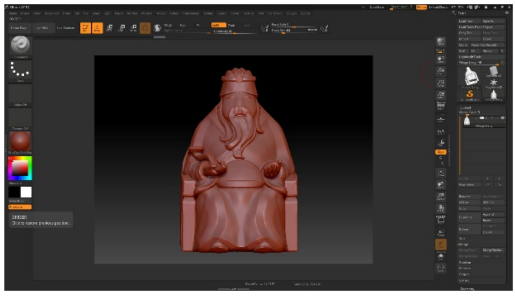

Fig. 3. ZBRUSH-digital model sculpture design

Fig. 4. SubstancePainter-material texture drawing

The inter-software conversion uses the common model format FBX. FBX files can contain lights, cameras, textures, animation, skeletal animation, material properties, polygons, curves, surfaces, normal maps, texture coordinates, and point group materials. The major 3D software is completely compatible with the Unreal game engine.

3.4 Motion Capture

For motion capturing, we adopted "Shōgun," a new motion capture software program that has been recreated by Oxford's Vicon Company in the United Kingdom. It improves the

quality of real-time motion capturing and also considerably shortens the postproduction processing time. After a 3D character model was constructed, the rigging was performed in Maya. We then used a free third-party software program, Advanced Skeleton, which is a fully functional rigging tool that has been continuously developed to support Maya updates. The Advanced Skeleton system can also retarget motion capture and facial expression capture data (Fig. 5).

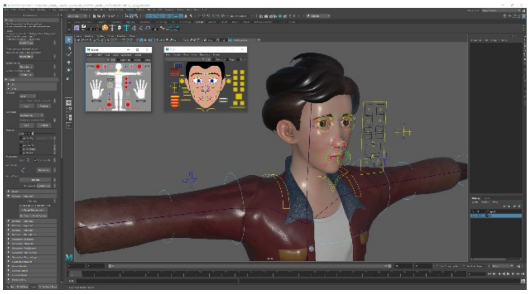

Fig. 5. Free third-party software (Advance Skeleton)

We hired a professional actor to engage in performance and used motion capture to record his performance as digital data. Next, we optimized the data using a post-animator and polished the performance (e.g., corrected action data generated by the recording). We also enhanced the subtle expressions and delicate muscle changes of the characters' fingers—particularly important when showing idol craftsmanship—to authentically present fine animation and dramatic acting (Fig. 6).

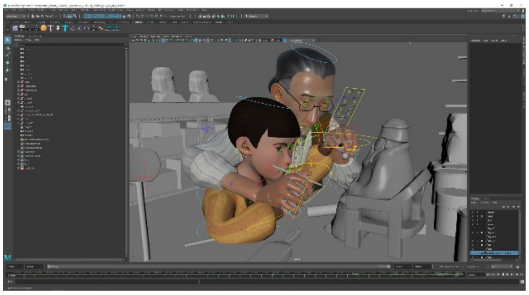

Fig. 6. Hand motion and finger gesture

3.5 Faceware

In the past, we had to manually adjust the facial expressions of animated characters, which can produce stiff and unnatural results. Here, we used Faceware to accurately capture facial performance and preserve subtle expressions. During his performance,

the actor wore a homemade helmet with a camera attached to the front to fully capture and record his facial performance. Subsequently, we used Faceware Analyzer to analyze the recorded video, set the tracking point, and save the data as an FWT file. After the 3D character digital data were imported by rigging into Maya, Faceware specified the corresponding end, effectors. An initial animated version of the character's facial expressions and movements was created, which was later optimized and finalized by the animator (Fig. 7).

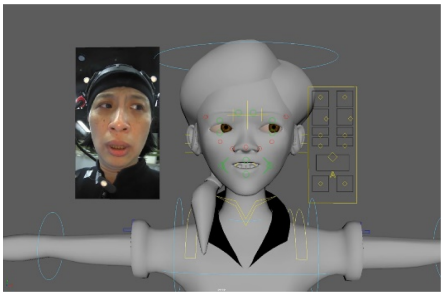

Fig. 7. Faceware effects

3.6 Integration in Unreal Game Engine

Construction of Virtual Space. We measured the 3D scene space in advance. According to our measurements, we limited the sensing area and placed furniture purposefully to avoid the use of oversized objects so as not to block the camera or cause mistakes. The more accurately virtual space is matched with real space, the more accurate action data are, which can considerably reduce adjustment time (Fig. 8 and 9).

Fig. 8. Scene space measurement

To unify the size of the Maya and Unreal environments, we first constructed all scene objects and imported them into Unreal as a blueprint. All the models were mapped

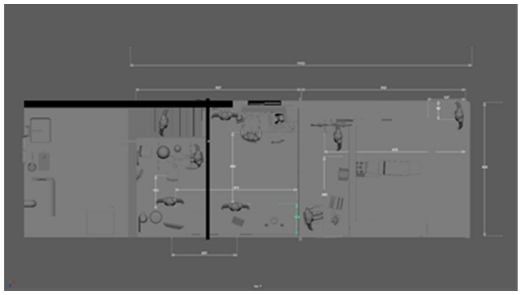

Fig. 9. Virtual space measurement

and completed in Substance Painter and then imported into Unreal (Fig. 10). All the texture files exported by Substance Painter were divided into BaseColor, Normal Map, and OcclusionRoughnessMetallic. OcclusionRoughnessMetallic is channeled in RGB, which are Occlusion, Roughness, and Metallic. These properties are synthesized as one texture to save memory space. Unreal, like Substance Painter, utilizes PBR. As long as the channel is matched, the display will be the same as in Substance Painter.

Next, we positioned the processed model material with Asset according to the blueprint. Pre-calculating the lighting effects into LightMap can considerably reduce the lighting-related computing resources required, although instant rendering also allows the animator to easily adjust the lighting in the Unreal environment. With this GUI functioning of the WYSIWYG process, the animator can focus on aesthetic design to improve the graphics quality without waiting for rendering.

Pistage Application. Maya has always been better than Unreal in the operation of action design and action capture data, so we usually process animation characters in Maya and then test the converted FBX files in Unreal. Any problematic files are transferred back to Maya for correction. However, this is not efficient at all. Therefore, the plugin tool Pistage in Maya helps connect Maya and Unreal. While specified files are being edited in Pistage, the adjusted and modified motions are instantly transferred to the file in Unreal. In other words, the final results of lighting and parameter configuration can be instantaneously previewed, which greatly shortens the conversing time.

To ensure that digital characters accurately interact with desired objects, we built the scene and set a camera in Unreal and then imported the converted FBX files to Maya. The characters were constructed as 3D models in a 3D environment. At this stage, we adjusted the characters according to scene planning and lens arrangement and also polished action details and modified their interactions with scene objects.

The animation of a character's emotions and facial expressions was tested in Unreal to determine if an adjustment of the lighting would be required. The animation was then modified using the Unreal particle system and reinforced and embellished with special effects.

VR360 Sequence Rendering. Unreal is a game engine wherein it is common to view ambient objects in 360°. The program is efficient at capturing 360° continuous images through a camera with instantaneous rendering. The rendering of an image with 3840

× 1920 resolution on a display card (GTX1080) takes an average of 1 s. A cube is photographed by an Unreal internal camera with a 360° planar acquisition system. Each facet of six sides of the cube is captured at the 2-K resolution, after which all the facets are tied together and projected onto a sphere and then deformed and converged to obtain the final full image (Fig. 10).

Fig. 10. Final result of VR360

4 Methodology

To optimize our production process, we interviewed the production team leader and three technology artists. The following interview questions were asked:

1. Regarding the production process and technical aspects, please compare the differences between VR360 animation production and conventional animation production.
2. What software and hardware are used in the production process and what are their respective functions?
3. How do you seamlessly integrate different software programs?
4. Please state the problems you face and the solutions you employ regarding your job content and tasks (such as concerning animation, lighting, and motion capture).
5. How do you operate a computer effectively in case of memory space and performance issues? For example, do you have a backup of material texture files?
6. Are there different requirements for VR animation and conventional animation in terms of resolution?

Our interviews included discussions of the establishment of a VR360 animation production process, the integration of different software programs, and problem-solving.

4.1 VR360 Animation Production Process

To obtain the best final product, we experimented, solved problems, and optimized the production process (see the flowchart in Fig. 11). An optimal VR360 production process

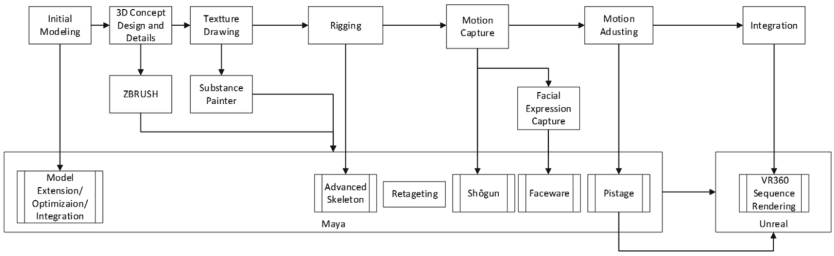

Fig. 11. Optimization of VR360 animation production process

contains seven stages, namely initial modeling, 3D concept design and details, texture drawing, rigging, motion capture, motion adjusting, and final integration.

The fundamental difference between conventional 2D/3D animation and VR360 animation is narrative techniques. Conventional animation uses the 2D shot as a basic unit, after which a series of frames on a single timeline is converted into a linear film. The audience views such a film and can understand its metaphor of a created story. The development of VR360 animation vastly supersedes this production. The immersive power is unique, with the audience being able to fully enter a scene as a protagonist and experience the story in a virtual environment. VR360 provides an innovative expression form that allows artists to explore and experiment with their narrative abilities.

Due to the advances in 3D and photographic technologies, VR360 animation framing is remarkably different from conventional animation with its single-plane composition. To depict a scene, no dead angle should be included. Such a concept is borrowed from conventional animation. For VR360 animation, the distortion projection must be calculated by a computer and captured by a 3D camera. Based on human knowledge related to reality, 3D scenes and characters must be constructed in a 3D environment and realistic VR360 images, which is a technical threshold that cannot be met by the techniques of hand-drawn 2D animation.

4.2 Integration of Different Software Programs

Maya, Substance Painter, and Unreal were the main software used in this production. Maya was supported by third party plugins such as Faceware and Pistage. The primary advantage of Unreal was the instantaneous rendering, which enabled images to be previewed instantly. Unreal is also adequate for editing and adjusting. Its fast calculations work effectively and shorten overall production time. Regarding resolution, due to the 360° rendering technology demanding a better result, we increased the resolution from 4K to 8K and conducted several tests to achieve the optimal effect and efficiency.

Faceware (Face Dynamic Capture System) can capture the live performance of facial expressions. The captured motion is vivid and makes it easy to synch a live speaking mouth to an animated character.

Substance Painter provides a more intuitive method of drawing material textures than does Photoshop or 3Dcoat. Through the material process and instant preview, artists can create new material textures in a short time with a few simple configurations.

We are a small team of Maya users. In terms of interacting with interfaces, Unreal was new to us all. It was not efficient to make everyone learn to use Unreal, but Unreal results can be previewed in real-time and we were able to create animations more intuitively. Pistage connects traditional and modern animators, allowing us to preserve old production habits while also familiarizing ourselves with new techniques. This enabled us to focus on aesthetic effects and artistic expressions.

4.3 Problem Solving

Problems occurring in the production process were largely technical issues, such as motion capture, lighting, and memory space. We created a cartoon-like character and needed to track motions repeatedly to ensure that this character stylishly would not inconsistent with other characters, as well as optimize its artistic style matched with the entire film.

Though motion capture systems performed most of the work for us, the results were heavily modified. Certain motions could not be captured by the motion capture system, such as minute finger motions; thus, we were required to create an entirely new animation that mimicked the actor's performance as best as possible. This was time-consuming.

The game engine does not produce better lighting. To increase efficiency, we precalculated and stored lighting information at the preproduction stage. It took more time than we expected; however, the advantage was that the action and material could be previewed instantaneously. Eventually, the rendering effect became more intuitive.

Generally, we used PBR rendering, but there are many channel textures (e.g., Base color, Roughness, Metallic, Normal, Op, and AO). Unreal integrates the channels of Roughness, Metallic, and AO into one material texture and controls these channels simultaneously. This allowed us to export the BaseColor, Normal, and three-channel textures from Substance Painter to Unreal. This advantage directly reduced the number of material texture files we had. In short, we were able to use memory space effectively. Unreal also greatly reduced the complexity of processing and shortened production time.

5 Conclusion

VR 3D animation technology is a suitable solution for dealing with the production of VR360 animation. Currently, advances in the development of production tools induce mature productions. The public's pursuit of high-quality images has promoted the progression of computing technology. Conventional full HD picture quality is insufficient to display spherical VR360 films. Although a 4-K quality picture is still considered above standard, to pursue a better movie experience, 8K or even 16K will undoubtedly become the future trend. Such a pursuit also aims to increase the threshold and restrictions on creative and experimental productions. Therefore, by borrowing technology from 3D game engines, using their real-time display rendering power, and instantly modifying WYSIWYG features, we considerably improved the VR360 production process and the final animated film result.

Unreal Engine 4 is the game engine released by Epic Games. It is the world's best-known and most widely licensed top-level game engine, accounting for 80% of the global

commercial game engine market. Many VR games are developed using this game engine. This is because Unreal has particular advantages in rendering effects and operational efficiency that are crucial for the VR experience. Recently, it has become more actively used in film and television productions. Unreal's excellent VR function and expansion capabilities make it easy to integrate 3D software, which has the advantages of plugin tools, to obtain high-quality VR360 images.

This was our first VR360 animation production, finalized by the Unreal game engine and aided by several types of 3D mainstream software. By using FBX files and PBR materials, we faced relatively few technical issues integrating these software programs. The instant preview feature of Unreal also made the error review process smoother. With high-speed rendering, it was effective at correcting errors, solving problems, and rerendering quickly. With 4K resolution, continuous images can be quickly captured and image presentation is satisfactory while the static or mirror movement is indivisible. However, for light or material details displayed in a large screen scene, it is easy to see flickers and jagged images. To solve these problems, we used 8K resolution for our highly animated scenes. The quality was improved, the jagged images could be completely dissolved. Due to the limitations of our hardware equipment, the final resolution was 3840 × 1920, with a single-eye resolution of 1440 × 1600 pixels (two eyes: 2880 × 1600 pixels). However, the final result viewed in the VR device is satisfactory. (The 47th International Conference and Exhibition on Computer Graphics & Interactive Techniques/SIGGRAPH, Computer Animation Festival Electronic Theater: Official Selection: https://s2020.siggraph.org/presenter/?uid=2082287891416180395).

Acknowledgement. This project is funded by the Kaohsiung Film Archive in Taiwan, ROC.

References

1. Ramos, L.A.G.T.: Production of 3D animated short films in Unity 5: Can game engines replace the traditional methods? Dissertation, Portuguese Catholic University, Lisbon, Portugal (2017)
2. Beane, A.: 3D Animation Essentials. Wiley, Indianapolis (2012)
3. Menache, A.: Understanding Motion Capture for Computer Animation. Morgan Kaufmann, Burlington (2011)
4. Gleicher, M.: Animation from observation: motion capture and motion editing. SIGGRAPH Comput. Graph. **33**(4), 51–54 (1999)
5. Shin, H.J., Lee, J., Shin, S.Y., Gleicher, M.: Computer puppetry: an importance-based approach. ACM Trans. Graph. **20**(2), 67–94 (2001)
6. Derakhshani, T.: Film Review - 'Rebels of the Neon God'. Philadelphia inquirer. World Wide Web (2015). https://www.inquirer.com/philly/entertainment/movies/20150703_Film_Review_-__Rebels_of_the_Neon_God_.html. Accessed 05 Aug 2020