

The Purpose of Maintaining Polycount of 3D Models In Animation Cinema And Video Games

Abhishek Bhattacharjee

Assistant Professor

Amity School of Fine Arts, Amity University, Kolkata

Abstract- *Until now, polygon count, also known as poly count, has been a basic component in both the technical and artistic aspects in terms of creating models for three-dimensional animation and video games. The role it plays in determining and judging the level of fidelity and rendering performance—not to mention the cost and usability aspects—is imperative in animation and video games. Although huge advances in both hardware and software technologies capable of handling geometric complexity, this aspect has played a fundamental and integral part in both animation and video games. In this academic research, a discussion on the importance and dimensions of ensuring and maintaining its optimal polygon count in models for animation and video games, and their ideologies, through a comparative analysis and discussion on differing programming processes in both aspects in relation to its importance in relation to its necessities in both aspects, shall be explained and defined.*

Keywords: Polycount, 3D modeling, animation cinema, video games, real-time rendering, optimization, computer graphics, performance efficiency.

I. INTRODUCTION

3D computer graphics have developed greatly in the entertainment sector, from cartoon animations to interactive 3D gaming environments. In the 3D graphics realm, 3D models are the primary foundation of digital storytelling. The geometric model with polygons that describe shape and form, and the boundary of the surface details forms the vital part of the nucleus of all 3D models. The total number of such polygons in a model, known as polycount, has a precise impact on the capability of handling and the visualization of the model itself. Polycounts, therefore, are a crucial step in the process of creating 3D assets. While increased poly counts would translate to models that are more aesthetically pleasing, the actuality of the matter is that it is much, much more complex. A poly count that is too high will easily result in speed issues and increased rendering times, including Animation and even increased production costs. A model, on the other hand, with reduced poly counts may very well lack sufficient detail for the creation of realism. This parity can become ever more important in relation to the explanation of

animation cinema and video games side by side. Though animation cinema and video games both use 3D models and analogous tools in animation software, they can be found to be vastly different from each other in terms of their technological requirements and creative objectives. Animation cinema continues to be based on offline rendering methods wherein images are rendered prior to the real-time constraint necessities of video games in a self-supporting manner that instantly satisfies their rendering requirements based on player input in video games as per their respective controls. Conversely, video games are totally under the control of real-time rendering methods wherein images are computed instantly based on fundamental player inputs in video games. Therefore, analogous polycount handling methods in animation cinema and video games become necessary based on their respective control requirements and different objectives in computer graphics. In general, it can therefore be stated that a significant purpose in this research paper would be the objective explanation and necessary justification of why polycount in relation to 3D models must be preserved in animation cinema and video games.

Understanding Polycount in 3D Modeling

Polycount is the reference of the number of polygons in a 3D model. Each polygon refers to a flat surface, whether a triangle, a quadrilateral, or more. The reason polycount is always referenced in terms of triangles is that the computer today is working purely with triangles in terms of calculations. Polygons relate to complex shapes or surfaces. Complex shapes or surfaces are described with curves in computer graphics. Many polygons refer to high detail, while few mean either a sharp or crude representation or object. One must consider, however, the following in computer modeling in terms of polycount: Polycount does not relate to the picture's accuracy or picture or representation. Methods or procedures in terms of the following:

Textures/Mappings/Displacement/Shaders proves that one can provide a high detail picture without having a high polycount or a likely displacement or distortion in terms of costing or modeling.

Historical Context of Polycount Limitations

During the initial stages of 3D graphics, polycount was extremely limited due to hardware constraints. The number of polygons in video games and animated films was no more than thousands or low hundreds due to the limited power of processors and the small size of their memory. Although polycount was necessary due to constraints on hardware performance, as soon as technology advanced with increased power from processors, graphics cards, and other APIs, polys soared to new heights. Millions of polygons made their way into animated films, and video games boasted better characters and environments with increased polys. However, instead of eliminating polycount as a factor, it imposed new needs with regard to graphic detail. Even with such powerful technology, polycount remains an essential consideration due to increased complexity, high-resolution monitors, lighting effects, physics, and games on multiple platforms. The animated movie uses offline rendering, where the rendering of the movie takes place frame by frame without needing communication between frames. Though it means that a movie can have higher polys than a game, the time it takes to carry out the rendering makes the high poly model take several hours per frame. A complete animation movie is made up of tens of thousands of images. Small polycount inefficiencies can quickly increase the time spent rendering the images. A good polycount can guarantee the effective use of the rendering farm. Animation movies have highly populated scenes with characters, environments, props, and special effects. High polycount values lead to storage overflow, which reduces the loading speed of the particular scene, thus adding to the problems in special effects by lighting and rendering. Therefore, proper control of polycount ensures that the scenes will not be populated and will work well at all stages of animation. In animation movies, the characters require highly advanced systems in terms of animation and expressions. Highly populated geometries tend to make the skinning process complex, and it is difficult to animate. Proper topology with proper polycount enables easy deformation and animation. All animation productions do not target hyper-realistic productions. A large number of these productions use stylized visual approaches, the benefit of which comes from simplified geometry. This ensures that the poly count remains at the discretion of the artists as they try to maintain an artistic intent with respect to form, silhouettes, and animation as opposed to the attention to detail that the vertices entail. Animated film productions usually entail the departments of modeling, texturing, rigging, animation, lighting, and rendering.

Purpose of Maintaining Polycount in Video Games

Computer games need to render these images at speeds ranging from 30 frames per second to 60 frames per second or even faster, which basically means the image is displayed on the scene for some number of milliseconds. Too many polys just clog the GPU and can stall the fps lock. This was another factor that had to be considered to reach stable frame rates. Dependability in terms of gaming performance is yet another imperative issue toward attaining gaming engagement. Games are developed on varied platforms, ranging from highly advanced PCs to game consoles, mobile phones, etc. Each has the capability to improve in different ways. Optimization in the capability of different platforms can be assisted by Polycount Optimization. Today's games can have great open environments with a high number of assets that are rendered at once. While the polycount of one single asset may be within an acceptable range, the geometric cost due to all of them together can be substantial. Strict polycount budgets allow us to create lush environments without having to pay a performance penalty. Unlike cartoons, computer games require an immediate response to any kind of input that has been entered by the gamer controlling the game by means of any kind of control. The higher the poly count, the longer the input, thus posing issues for playability within computer games. A balanced geometry would guarantee that maximum responsiveness at the level of interaction between the gamer controlling the computer game would be attained. While gaming on a network, issues pertaining to performance as well as synchronization not only exist for the gamer, but they exist for the network as well.

Comparative Analysis: Animation Cinema vs. Games

Offline Rendering Versus Real-Time Rendering

The biggest difference between animation cinema and games has to do with the methodology used in rendering the image. Animation cinema uses offline rendering, enabling higher polycounts per asset, while games use real-time rendering with severe limitations.

Predictable Versus Dynamic Camera Systems

In animation movies, the camera movement angles are fixed, which helps the polycount to be allocated accordingly. In games, the camera movement angles are typically controlled by the gamer, meaning the polycount, or the objects, would need to be designed to account for every angle they would be facing at the same time.

Level of Detail (LOD) Usage

Most modern games are highly dependent on LOD techniques, which automatically change the level of detail based on the distance of the object from the camera or its significance to the environment. In situations where the object moves away to a considerable distance, there is a swap in the engine from the high-poly models to low ones, possibly with low-resolution textures or even billboards. This process must be seamless to the eye but also minimize the processing strain on the GPU and CPU, thereby making frame rates stable and game play responsive in real time. While LODs can also be applied in animation and filmmaking, because of the interactive and real-time nature of gaming, the importance of LODs during game development cannot be underrated. Speaking of film production or pre-rendered animation, it is important to point out that rendering the scene is an offline process that allows the artist to work on very high resolutions without concerning themselves with rendering in real time. The fact of the matter is that, in terms of video games, it must be able to render one frame from the playing screen of the entire game in real time, which is above 60 frames per second. Furthermore, the current generation of games has massive open worlds where dozens, if not hundreds, of different objects can potentially be viewed simultaneously in the same area. This makes the employment in the scene of the concept of priority-based model details, in which details would first be considered in the location that would be the most visible, that is, in the vicinity that the player's perspective is in, aggressive mesh simplifications, possible because of the use of the LOD methods.

Interactivity Versus Visual Control

Games favor interactivity and real-time rendering, which requires the display of assets smoothly as they react immediately to user input. This leads to a conservative handling of polycount in favor of polycount rendering in important aspects of the play action, often avoiding this in other aspects through level of detail and normal mapping. On the other hand, animated cinema emphasizes the display of artwork through cinematic animation. Since animation is performed offline, the use of high polycount becomes predominant in the creation of animation. This leads to a situation where the game favors illusion of functionality, and the animated cinema favors artwork.

Techniques for Maintaining and Optimizing Polycount

a) Retopology

The reconstruction process that ensures the creation of an optimal mesh with high resolution on the sculpture's surface, retopology aims at ensuring that the polygon count on

the model that needs to be created is minimized. Retopology helps in ensuring that deformation, handling, and rendering are easier on the sculpture's surface. According to the topic, it does not compromise on the look.

b) Normal and Displacement Mapping

Normal maps are used to create an illusion of details without adding geometry, which makes these maps so important in the context of low-poly games. Displacement maps, on the other hand, are mostly employed in animation movies, where the details are created through rendering without increasing polycount.

c) Level of Detail (LOD) Models

For performance optimization in games and big scenes, several versions of a model with decreasing polycounts are created.

d) Modular and Instanced Assets

Recycled optimized resources help save overall polycount memory. A modular approach to environment creation is especially beneficial for games.

e) Culling and Optimization of Visibility

Frustum culling, occlusion culling, and back face culling can be used to preclude the rendering of unrendered regions, thus improving performance without affecting the quality of the rendered image.

Artistic and Narrative Impact of Polycount Management

A good poly count also serves to retain good silhouettes and readability. Too much detail contributes to an obscurity of form and readability, especially when in fast-moving environments or gameplay. Low and balanced poly count results in smooth animations and deformations, enhancing the ability of the character to perform. Low-poly-count games or motion pictures can be very successful and can contribute to their success artistically impressive despite their geometric simplicity.

Future Trends and Ongoing Importance of Polycount

Such new technologies as real-time ray tracing and virtual geometry raise the chances of rendering, but it does not counter the limitation of performance in any way. Geometry optimization remains even more in-demand. The overlapping of the movie and gaming industries causes an even bigger need to know the things of polycount. Optimized polycount

results in an optimized reduction in the usage of energy, and the hardware degradation is also optimized.

II. CONCLUSION

Polycount of 3D models is one of the least, but less significant, tasks in animation movies and computer games, even though polycount remains one of the most fundamental necessities in these two different concepts. Polycount refers to the number of polys used in the generation of 3D models. This has an immediate impact on the ensuing result in the larger digital environment. Although animation movies and computer games are quite different concepts in terms of the handling of images, they share quite a number of fundamental necessities in the context of managing polycount. In animation movies, polycount is directly related to quality and realism. Films can be pre-rendered, and artists can make very detailed characters, environments, and props with a higher amount of polygons. However, even in this context, polycount management is essential. Excessively high polycounts will increase rendering times, storage requirements, and production costs by many factors. Studios need to find a good balance between obtaining high visual fidelity and efficiency in their productions by getting polygons to the right locations—for example, facial features or locations nearest to the camera—while using less detail elsewhere that is less noticeable. In animation movies, polycount is directly related to quality and realism. Films can be pre-rendered, and artists can make very detailed characters, environments, and props with a higher amount of polygons. However, even in this context, polycount management is essential. Excessively high polycounts will increase rendering times, storage requirements, and production costs by many factors. Studios need to find a good balance between obtaining high visual fidelity and efficiency in their productions by getting polygons to the right locations—for example, facial features or locations nearest to the camera—while using less detail elsewhere that is less noticeable. In video game development, polycount is even more essential since everything is rendered in real time. High polycount models can be associated with performance issues when rendering the game because the game can lag or freeze. Therefore, polycount optimization is even more essential, as the primary goal is to enable the game to meet the critical milestones of smooth game performance, fast loading times, and hardware independence. This is achieved using various methods such as the level-of-details models, normal mapping, and topology models. Despite the rapidly progressive technology revolution that has been observed in modern graphics cards and rendering engines, poly count remains a significant element. Contrary to the assumption that its significance would decrease, it has instead matured to become an emerging new critical element in 3D graphics production.

Today, modern artists are more required in modern history, being tasked with making critical decisions that include but are not limited to their artistic judgment against technological constraint.

REFERENCES

- [1] Catmull, E. (2014). *Creativity, Inc.: Overcoming the unseen forces that stand in the way of true inspiration*. Random House.
- [2] Eberly, D. H. (2006). *3D game engine design: A practical approach to real-time computer graphics* (2nd ed.). Morgan Kaufmann.
- [3] Kerlow, I. V. (2009). *The art of 3D computer animation and effects* (4th ed.). Wiley.
- [4] Lengyel, E. (2012). *Mathematics for 3D game programming and computer graphics* (3rd ed.). Course Technology.
- [5] McGuire, M., & Luebke, D. (2010). *Hardware-accelerated global illumination*. AK Peters.
- [6] Möller, T., Haines, E., & Hoffman, N. (2018). *Real-time rendering* (4th ed.). CRC Press.
- [7] Parent, R. (2012). *Computer animation: Algorithms and techniques* (3rd ed.). Morgan Kaufmann.
- [8] Pharr, M., Jakob, W., & Humphreys, G. (2016). *Physically based rendering: From theory to implementation* (3rd ed.). Morgan Kaufmann.
- [9] Unity Technologies. (2020). *Optimizing graphics performance*. <https://docs.unity3d.com>
- [10] Unreal Engine Documentation. (2021). *Performance guidelines for artists and designers*. Epic Games. <https://docs.unrealengine.com>
- [11] Williams, L. (1983). Pyramidal parametrics. *ACM SIGGRAPH Computer Graphics*, 17(3), 1–11. <https://doi.org/10.1145/964967.801126>