Emerging Trends and Technologies in Graphics Rendering Pipeline

[1]Priyanshu Bhattacharjee
Department of Computer Science and Engineering
Chandigarh University
Gharuan, India
priyanshubh999@gmail.com

[2]Sidharth Kumar
Department of Computer Science and Engineering
Chandigarh University
Gharuan, India
siddharthkr170@gmail.com

[3]Mausam Kumar
Department of Computer Science and Engineering
Chandigarh University
Gharuan, India
mausamkumarsingh70@gmail.com

[4]Geet Kiran Kaur
Department of Computer Science and Engineering
Chandigarh University
Gharuan, India
Geetkiran.bhabra@gmail.com

[5]Ankul Kumar
Department of Computer Science and Engineering
Chandigarh University
Gharuan, India
ankulmuz666@gmail.com

***Abstract*—The Graphics Pipeline enables the rendering of 2D and 3D images on various output devices, including computer monitors, mobile screens, and VR headsets. It is generally used in the Graphics Processing Unit (GPU). A graphic rendering pipeline includes stages of Application, Input Assembly, Shader Vertex, Tessellation, Shader Geometry, Rasterization, Shader Fragment, Depth & Stencil Testing, Blending Process, and Output Merger. The Graphics rendering pipeline is a main component in graphics systems that enable real-time rendering in gaming, VR, simulation, film production, and various visualization production. Current Graphics APIs including Metal, DirectX, OpenGL, & Vulkan are essential tools for providing access to GPU hardware for rendering 2D and 3d graphics efficiently. These APIs enable advanced graphics techniques for high-performance rendering and better resource management. It constantly changing to face the increasing need for more engaging and realistic visuals in different fields such as gaming, VR, film, etc. This evolution aims to enhance user experiences and improve visual quality.**

Keywords: Graphics Pipeline, Software Rendering, GPU, Rasterization, APIs

I. INTRODUCTION

Computer graphics is a quickly developing field in computer science, indeed despite it has only been around for two or three decades. Since most people handle information visually, out all sectors- including science, industry, and entertainment depend on Computer graphics. This depends on making graphics a key center in both research and industrial development inside computer science [1][2].

Creating a 2D image in Computer Graphics from a 3D scene is called Rendering. On the other side, converse rendering is the inverse: it reproduces a 3D scene using various 2D images as input [6]. GPUs support both Inverse Rendering & Traditional Rendering using well-defined software tools [3]. They are used in all kinds of applications, including video games, movies, visual presentations, and 3D images. These strategies usually have a few challenges. They regularly require manual input from talented experts, and the specialized software can be complicated. This complexity can make it harder for people in different industries to create and visualize 3D content efficiently [7]. Some rendering systems prioritize using the specialized hardware pipelines that are discovered in GPUs like Direct3D and OpenGL pipelines. Others depend on computer-based pipelines which manage CPUs or take benefit of the configurable features of contemporary GPUs [5]. We focus on the challenge of generating a pipeline for graphics that is extremely useful for programming & performs properly with the help of utilizing the concurrent processing abilities of GPUs. While the development of the graphics pipeline moves gradually, the computing power of GPUs increases exponentially [4].

Modern APIs such as Vulkan, OpenGL, or Direct3D, as well as specialized tools like CUDA and OpenCL also allow GPUs to operate in compute mode, that shows GPUs can act as general-purpose processors using multiple cores that operate at the same time or work in parallel [8]. For real-time rendering, the hardware graphics pipeline remains central, & advanced graphics rely on compute mode to handle complex tasks like deferred rendering, advanced shading, or cloth simulation which are challenging to accomplish only with the traditional graphics pipeline [16].

Methods of General-Purpose Graphics Processing Units (GPGPUs), have risen to importance rapidly because of potential unusual parallel features in Graphics shaders. Large-scale parallel processing skills are now available even in general-purpose computations with the execution of modern parallel processing and frameworks such as CUDA and OpenCL [8]. Over the years, the graphics pipeline has been involved in a process similar to precipitation: Chaotic in nature and the number of higher-level software implementations, which are considered for standardization and integration into the pipeline. This process is ongoing and, in this article, we will review recent changes introduced into the pipeline while also presenting different perspectives for pipeline improvement [9].

The render's APIs allow just personalized shader implementation along with some limitations (such as 3D visuals). The above renders are understood as fixed or closed. Aside from proprietary coding variables, the primary challenge is to influence the status of high-end rendering pipelines, these days is a huge need for rapid photorealism or perception-based rendering [9][11][12]. This results in a small number of methods in the field of specialized research, but it also promotes the development of traditional approaches [10].

The integration of media objects has provided birth to a variety of creative frameworks, particularly Berkeley Vector-IRAM, Standford Imagine, & MAP3D. Vector processing and incorporated methods are utilized in the Berkeley Vector-

IRAM vector system [13]. With pipeline design for short vector creation, high memory bandwidth, and flexible and quick support for various data types, it boosts up media processing. Eight working clusters, a 128-kbyte streaming record file, & a four-channel SDRAM memory system constitute elements of the Standford Imagine. It allows to perform referred to as stream programming, which splits positions into several basic computations. In place of memory, this programming method generates an enormous stream record file containing the input and output data for each center estimation. To achieve high-performance video, image, & 3D graphics processing, MAP3D is a combined and 3D graphics VLIW media processor and a 3D graphics pipeline which make best advantage of the latest gadgets [14]

Table 1: Basic comparison between Traditional 3D Rendering and Modern 3D Rendering

Features	Traditional 3D Rendering	Modern 3D Rendering
Graphics Quality	Limited resolution, color depth, basic textures, and flat shading	High-resolution textures, bump mapping, advanced lighting techniques
Rendering Techniques	Rasterization with basic shading (flat and gouraud shading) and lack of techniques for realism	Rasterization and ray tracing for realistic images and shading techniques (Phong shading, screen space reflection, ambient occlusion
Hardware and Performance	Basic hardware, with limited processing power (older CPUs and GPUs)	Powerful GPUs and parallel processing capabilities, Real-time Rendering
Software and Tools	An early version of 3D modeling software, manual & time-consuming processing	Sophisticated software and tools (Blender, Maya, Unreal Engine), Automated features
Interactivity and User Experience	Mostly static scenes or simple animations: interactivity was minimal	Highly interactive environments, especially in gaming and VR, Dynamic content generation

Nowadays, shading is rather complex in real-time graphics. The other part of the graphics pipeline, which involves rasterization and triangle setup, is quite small in reverse, but an increasing amount of GPU resources moves into complex shader and texture components [18][19]. If an accumulation buffer or stochastic rasterization is implemented, the shading cost will increase linearly with the number of samples, rendering effect includes motion and blur that require significant sampling over a 5D domain (pixel area, shutter interval, & lens aperture) very expensive [13].

Therefore, for real-time applications, this effect frequently had to be approximated via predictions. Although shading frequently does not alert greatly throughout the shutter interval or lens perspectives, it can enhance on, high-quality motion and blur does require various samples of the visibility function over a 5D field [14].

This paper reviews ways to increase the fragment shading stage of the graphics pipeline by including techniques that sample portions of the shading function more lightly than per-pixel rates, enabling high-quality shading at a cheaper price on GPUs. A PC-Cluster parallel rendering solution termed Parallel SG implements a hybrid sort-first and sort-last framework [12]. Frame-to-frame duplication can be recognized in a TBR GPU at the tile level, so it's much earlier in the pipeline than it can be using other approaches. A simple example of how rendering Exclusion, an algorithm for early tile elimination, may be carried out into the Graphics Pipeline with little hardware and performance outlays follows via an explanation of the suggestion [10][19]. With the support of the head/eye tracking data collected at the beginning of the frame using the mobile VR render system, the app or game engine produces images for each eye as they happen at its own pace.

We review modifications to the OpenGL graphics pipeline which allows rendering to multiple perspectives in a single pass. Our way is to make MVR a conclusion (as opposed to a specialization) of SVR [15]. A new method to conquer the disadvantages of the Holoimage squeezing method by deleting its geometric encoding requirement. Alternatively, this method directly converts depth into RGB images. By employing an advanced computer graphics pipeline (such as OpenGL), this technique automatically converts 3D range data into typical 2D images. Through the computer graphics rendering pipeline (CGRP), 3D shape data can be converted into 2D images for displaying 3D shapes on a computer display [15][17].

II. LITERATURE REVIEW

In recent years, the domain of computer graphics has introduced many changes, it's become an essential component in various sectors including gaming, science, industry, VR, and entertainment. The graphics rendering pipeline, which converts 3D scenes into 2D images, is a main component of this Paper [18]. Rasterization and simple shading methods are used in the Traditional rendering, which results in low visual realism and interactivity. However, Ray tracing & physically based rendering are used in Modern rendering, which increases the visual quality and user experience [2][6][13].

Improvements in hardware and the development of powerful GPUs have innovated the rendering process. Present GPUs support real-time rendering, sophisticated lighting effects, and complex shading, and enable high-resolution textures [9]. APIs such as OpenGL, Direct3D, and Vulkan make the graphics pipeline simpler to program, enabling a more flexible and interactive framework. For efficient parallel processing, we focused on improving the features of the graphics pipeline which led to the rise of General-Purpose Graphics Processing Units (GPGPU) [14]. OpenCL and CUDA frameworks made it possible to utilize GPUs for computations, which improved the quality of rendering work.

Stanford Imagine and Berkeley Vector-IRAM are new rendering techniques, that help improve data handling and media processing with the help of Advanced System. Instead of these advancements, photorealism and simplifying the rendering process across gaming, industries have main challenges [8]. Modern techniques, including Physically-Based Rendering, Ray Tracing, Deferred Rendering, etc. continue to improve the efficiency and quality of computer graphics [15][20]. As we know graphics pipelines play a major role in software industries, we will review focusing on improving integration and standardization to face the increasing demand for visual experiences [20].

Fig. 1. Indicates the no. of papers we go through and researched of a particular year.

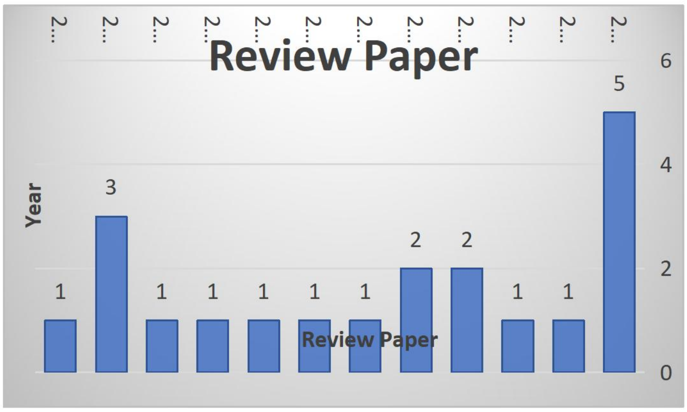

Fig.1: Reviewed paper from 2002-2024

Real-time rendering techniques like CLAY, DreamMat, and Ray Tracing help to understand deep learning with the help of GANs and Monte Carlo Sampling [17][19]. Photorealism, real-time rendering, and efficiency are improved by new rendering technologies [16]. Complex materials, consistency in lighting, and ambiguities in text inputs include many challenges [18][20]. Real-time ray tracing development focused on new algorithms that maintain performance across various hardware and applications [16].

Table 2: *Shows the Review of Research Papers on Graphics Pipelines for the GPU.*
Table 3: *Shows the Review of Research Papers on Graphics Pipelines for 3D Structures.*
Table 4: *Shows the Review of Research Papers on Graphics Pipelines for Rendering Features.*
Table 5: *Shows the Review of Research Paper on Graphics Pipelines for Grid Computing and Deep Learning in Rendering.*
Table 6: *Shows the Review of Research Paper on Graphics Pipelines for Mobile VR*
Table 7: *Shows the Review of Research Paper on Graphics Pipelines Published Recently*

These review tables include author names, techniques used, key findings, challenges they faced, and the year in which it was published.

Table 2: *Review of Research Paper on Graphics Pipeline for the GPU*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2011	Jonathan Ragan-Kelley, Jiri Lehtinen, Jiawen Chen, Megan Doggett, Fredo Durand [1]	A Separated Sampling method allows individual optimization of several parts within the rendering pipeline. The method runs on modern GPU techniques to utilize parallel processing for improving rendering speed and efficiency.	They render efficiency and visual quality without adding a large computational burden. It improves in the field of rendered images and integrates smoothly with graphics systems. Their experiments show that by adjusting different stages separately, significant performance improvement can be obtained.	While implementing decoupled sampling, one main challenge is the complexity of integrating these features into the present graphics pipeline, which is now available in traditional techniques. Due to the increase in implementation complexity and disruption in workflow, developers face many difficulties.
2014	Yong He, Yan Gu, Kayvon Fatahalian [2]	Introduce Adaptive multi-rating shading techniques that upgrade the traditional graphics pipeline. Implement various shading methods around different parts of a frame, which maintain performance of high-visual quality.	Adaptive multi-rate shading currently improves rendering performance without surrendering image accuracy. The Method can achieve major speed-ups in frame costs, mainly in geometric complexity. This method allows us to create a visually rich environment with less computational costs.	Integrating adaptive shading into existing rendering pipelines. Managing various shading rates carefully without introducing antiquities. Ensure consistency with various hardware and software environments.
2015	Anjul Patney, Stanley Tzenge, Kerry A. Seitz, John D. Owens [3]	It is made to simplify the production and improvement of graphics pipelines. This technique provides a flexible way, allowing coders to construct personalized pipelines with increased performance and flexibility.	Power to enhance efficiency in activities and support different tasks with little complexity through a specialized language and integrated framework.	Balancing flexibility with performance enhancement and securing the framework is flexible to progressing hardware structures without sacrificing performance or functionality.
2018	Michale Kenzel, Bernhard Kerbl, Dieter Schmalstieg, Markus Steinberger [4]	Introduced a new Program-based pipeline for graphics in GPUs. Their structure utilizes general-purpose GPU computing for rendering, providing a performance similar to hardware-based pipelines.	It involves reaching major performance boosts across traditional methods, particularly in complex rendering activities and non-standard use cases.	Enhancing the pipeline for several GPU structures and adjusting flexibility with performance. Difficulty in managing APIs according to their methods differs from standard hardware-accelerated rendering skills.
2019	Ghazi Shakah, Mutasem Alkhasawneh, Victor Krasnaproshin, Dzmitry Mazouka [5]	Examines progress in graphics pipelines, highlighting enhancement and improvements. It evaluates algorithms for rendering performance and GPU application. Parallel processing techniques and shader upgrades.	It includes GPU structure study and computer-based improvements in rasterization and shading points. Findings the importance of efficient memory organization and data flow rate for obtaining high-performance graphics rendering processing.	Balancing algorithmic complexity with real-time requirements, specifically in complex settings. Scalability concerns and Hardware drawbacks in Gaining optimal parallelism and the exchanges between rendering speed and quality.

Table 3: *Review of Research Paper on Graphics Pipeline for 3D Structures*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2002	Chris Yonghwan Chung, Ravi Ashok Managuli, Yongmin Kim [6]	Shows different types of computing structures designed around a 3D graphics pipeline. The structure utilizes a scalable design, enabling flexibility in various rendering applications.	Progress in rendering speed and resource use compared to traditional structure. Parallel processing in reducing bottlenecks during different data handling.	Securing consistency with existing algorithms and controlling the complexity of structure. Difficulties in enhancing resource allocation and load balancing between various processing units.
2012	Song Zhang [7]	It uses techniques like geometry simplification, texture mapping, and level of detail methods to effectively encode 3D data while keeping high image quality.	Reduce data size without compromising rendering quality. Compression ratios of up to 90% while retaining vital details. This method balances speed and accuracy for various applications like remote sensing and VR.	Securing consistency with existing rendering algorithms and controlling computational burden at the time of decompression. Complex geometries require further upgrading for better performance around diverse datasets.
2020	Mingyu Kim, Nakhoon Baek [8]	Hardware compatibility bound based on the CUDA system, it runs on various devices like GPUs, CPUs, and FPGAs with the help of OpenCL cross-platform nature. A Top-down approach is used in this research paper for rendering and pixel precision.	OpenCL helps to implement full 3D graphics without the help of graphics hardware. It is important for incomplete environment 3D graphics support. They make it possible to offer 3D features around a larger range of systems.	The complexity of tuning performance is supported by OpenCL hardware for different platforms with the help of hardware systems like CUDA. The main challenge is performance upgrading in parallel environments and balancing workload among different devices.

Table 4: *Review of Research Paper on Graphics Pipeline for Rendering Features*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2007	Tomas Agoston, Csaba Csuprai, Juraj Onderik, Roman Durikovic [9]	Pipeline for Modular Rendering created for flexibility and efficiency. The Component-based structure is used for quick coordination rendering techniques such as forward and deferred rendering.	They include significant performance for maintaining high-quality output. They provide a platform for rapid prototyping of new rendering techniques for developing computer graphics.	Increase performance while maintaining flexibility needs management of data flow and resource allocation. Increasing complexity and tests for module interaction pose challenges for developers.
2007	Aravind Kalaiarasi, Tolga Capin [10]	It is specially designed for autostereoscopic displays, which give a 3D experience without using special glasses. Interpolation view and depth buffering techniques are optimized for better performance.	They show real-time rendering for autostereoscopic displays in various applications like VR and gaming. A smooth visual transition between views is implemented and motion sickness or image distortion issues are resolved.	Rendering multiple views at a time is the main challenge. Bottleneck performance like complex scenes and dynamic processing requires more intensive computation.
2008	Haoyu Peng, Hua Xiong, Zhen Liu, Jiaoying Shi [11]	The hybrid structure technique is used for both rendering techniques sort-first and sort-last. Introduced dynamic rendering team (DRT), formed with multiple PCs to render large datasets on covered display walls.	Rendering Composing Display Pipeline (RCD) is optimized with DRTs for better load distribution and load efficiency. Smoother performance, and interactive, large-scale rendering workloads are managed by this system.	Issues of computation and latency in maintaining synchronization across parallel pipelines without damaging the real-time rendering process. Balance nested pipeline system and their loads.
2018	Marti Anglada, Enrique de Lucas, Joan-Manuel Parcerisa, Juan L. Aragon, Antonio Gonzalez, Pedro Marcuello [12]	Tile-based rendering technique is used in this paper which finds and removes redundant tiles from the graphics pipeline. Tile hashing and framebuffer are used to check further the rendering process and reduce memory bandwidth usage.	Early elimination of redundant tiles helps in improving rendering efficiency. Reduce 50% of the unusual computations in a graphics pipeline. Energy consumption decreases and performance increases by discarding tiles in embedded systems and mobiles.	The early detection and computational efficiency balancing process is a little difficult. An incorrect algorithm damages the performance gains and makes it difficult to improve the finding algorithm.

Table 5: *Review of Research Paper on Graphics Pipeline for Grid Computing and Deep Learning in Rendering*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2007	Ade Jonathan Fewings, Nigel William John [13]	Computational grids and graphics-intensive applications are combined with the help of computational grids. Data partitioning and pipeline streaming techniques are used for rendering among several nodes in the grid.	Performance and scalability increased by distributing graphics pipelines across grid nodes. Decrease latency while increasing communication between nodes.	The main challenges like data synchronization, load balancing, and network latency. Rendering quality and real-time applications while providing computational loads. Delays in smooth communication between nodes are difficult.

2021	Mark Wesley Harris, Sudhanshu Kumar Semwal [14]	They presented an Advanced Deep Learning Graphics Pipeline (ADLGP). It increases various deep learning stages to streamline rendering. Rendering 3D scenes into rendered visuals is done by Generative Adversarial Networks (GANs).	Accelerate the rendering process with the help of ADLGP by providing frames of basic 3D scenes. GANs deep learning model is used for faster and more stable rendering of scenes.	Training stability and computational sensitivity in GAN-based structures. Take careful steps while managing sensitivities across pitfalls in deep learning. High-resolution image generation is also the main challenge.

***Table 6:** Review of Research Paper on Graphics Pipeline for Mobile VR*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2019	Haomiao Jiang, Behnam Bastani, Rohit Rao Padebettu, Kazuki Sakamoto [15]	Low-latency ML was introduced by them in the graphics pipeline for mobile VR. Scene understanding, adaptive quality adjustment, and predictive rendering are some tasks by ML based on user interactions.	Improvement in rendering efficiency and latency reduction. It predicts user behavior for a smooth VR experience with the help of ML techniques. Faster frame rates in the system by ML to increase mobile VR performance.	Some challenges like the installation of ML in mobile phones. A trade-off between model complexity and computational performance. In different views, strong model performance is not available in data management and training strategies.

***Table 7:** Review of Research Paper on Graphics Pipeline Published Recently*

YEAR	AUTHOR	TECHNIQUE	KEY FINDINGS	CHALLENGES
2024	Tizian Zeltner, Andrea Weidlich, Fabrice Rousselle, Jan Novak, Petrik Clarberg, Benedikt Bitterli [16]	Presents system-related appearance model for Real-time rendering. Compute reflectance and sampling directions by the network are decoded with the help of hierarchical textures. Shading frames like mesoscale effects and microfacet sampling help in rendering efficiency.	Offline methods help in complex real-time, high-fidelity rendering. Shows Order-of-magnitude speed increase as compared with traditional models. Film-level visual quality is maintained in games and live previews. A smooth visual transition between views is implemented and motion sickness or image distortion issues are resolved.	Scalability enhanced with different materials is the main challenge. Coherent and divergent execution patterns are difficult to perform. Handling of real-time path tracing with different properties. Difficult to manage anisotropic reflections and complex surface properties.
2024	Yuqing Zhang, Lei Yang, Yuan Liu, Zhiyu Xie, Zhongyuan Lui [17]	DreamMat improves traditional 2D transition for generating high-quality Physically Based Rendering (PBR). Geometry and light-aware models are combined to implement Monte Carlo Sampling and enhance realism with the help of removing useless shading effects in albedo.	Solved limitations of previous methods with incorrect material decomposition. Lighting and geometry conditions help in aligning generated materials. DreamMat increases rendering effects and visual quality in various applications such as film, gaming, and VR.	Baked-in Shading is the major challenge, this problem occurs when 2D transmission structures are recycled for component generation. Fine-tuning models for lighting effects are handled through this approach with consistency among different Lighting Environments.
2024	Longwen Zhang, Qiwei Qui, Haoran Jiang, Ziyu Wang [18]	To upgrade 3D asset generation CLAY enhances different structure that combines Transformer-based mechanisms with Generative Adversarial Networks (GANs)	CLAY improves the quality and diversity of generated 3D models. Full accuracy in translating text forms into 3D assets. Realism satisfaction is provided by various applications in gaming, film, and virtual reality (VR).	CLAY challenges in handling complex textual inputs and ambiguous that harm suboptimal outputs. Refinement in data curation to enhance model standardization performance in different domains.
2024	Qi wang, Hujun Bao, Yuchi Huo, Zhihua Zhong, Rui Wang [19]	Advanced rendering techniques are used for utilizing deep learning to increase efficiency and image quality. Generative Adversarial Networks (GANs) for texture generation and Convolutional Neural Networks (CNNs) for denoising are used for more realistic scenes and faster rendering times.	Highlights Deep Learning rendering techniques while improving photorealism and reducing computational overhead. Utilize deep learning for real-time applications and data-driven approaches in generating complex graphics and textures. This paper also includes fewer computational costs and less processing times.	Data Dependency Generalization and Interpretability of deep learning structures in rendering are the main challenges. Training networks on diverse datasets and robust model geometry are difficult. Need more standardized evaluation metrics for better performance.
2024	Yuan Tan, Liang Chao, Jian Cheng [20]	They discuss next-generation rendering techniques like real-time ray tracing and physically-based rendering (PBR). It also maintains high frame rates to reduce computational loads.	Modern hardware advancements like GPUs with ray-tracing cores are used in real-time applications. AI-driven upscaling methods are implemented to enhance performance, image quality, film applications, and VR.	Need more efficient algorithms to reduce computational costs. Difficult in managing rendering backward compatibility in game engines and performance around resource-intensive applications.

III. RESULTS AND DISCUSSION

The research paper review shows the latest advances in the graphics pipeline. We discuss the analysis of GPU rendering, 3D images, grid computing, real-time tracing, rendering features, and deep learning, while initial techniques such as hybrid structures and modular rendering indicate benefits in computing speed and accuracy. Hardware integration and scalability have issues with it. We found that program-based and adaptive multi-rate shading pipelines improve rendering performance without affecting quality. We face connection and optimization troubles. In recent years, researchers often used AI and deep learning models to enhance 3D structure creation and real-time rendering in GANs. These methods create high-quality images with some difficulties in integrating computing efficiency and hardware flexibility.

There are various issues with the algorithm and backward computations that are solved by recent techniques such as real-time ray tracing and physically-based rendering (PBR) in real-time applications. In the end, the study shows a huge improvement in rendering flexibility and visual representation. However, different platforms remain to raise issues in ensuring compatibility, scalability, and effective handling usage.

We show a graph in *Fig.2.* where the graphics rendering techniques used over the years for rendering 2D and 3D images on various output devices. Start with traditional techniques like vector and raster graphics. Now, progress has been made to modern techniques like AI-based rendering and path tracing with the help of DirectX, shader, OpenGL, and real-time ray tracing. Also shows improvement in devices from PCs and video game systems to mobile devices and VR headsets.

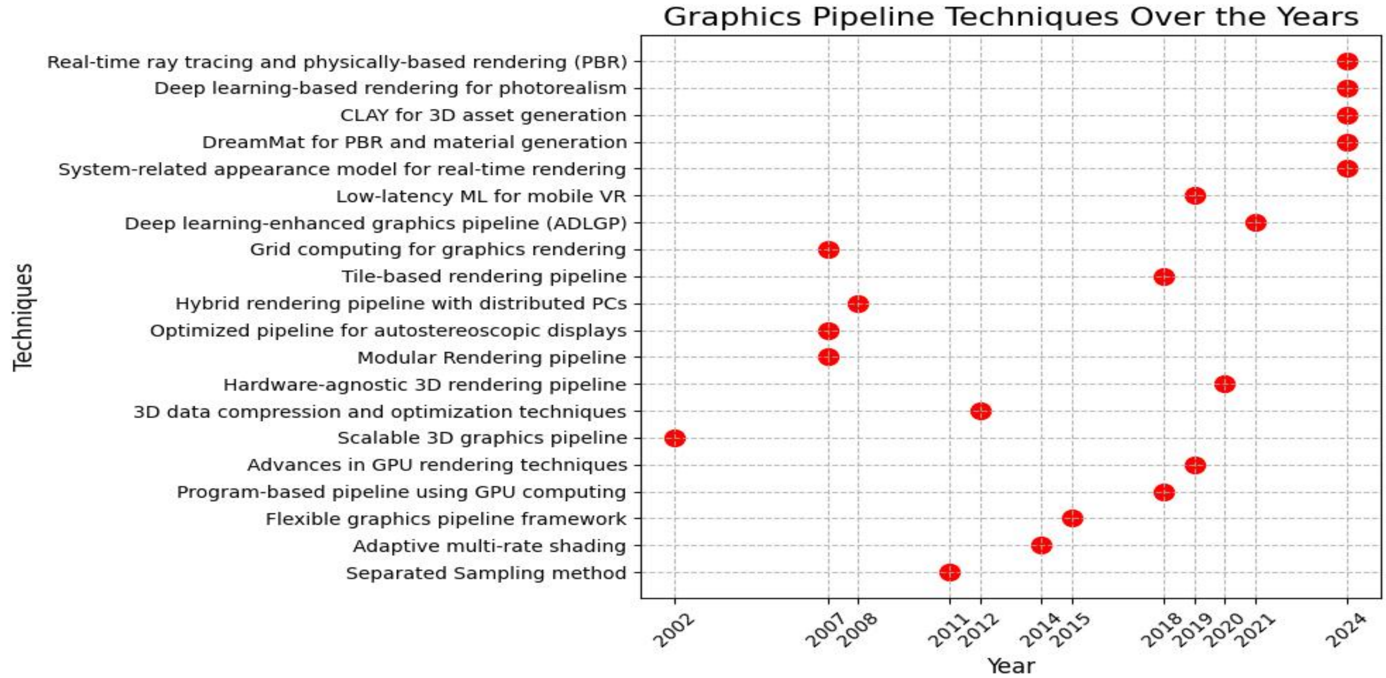

Fig.2: Graphics Pipeline Techniques Over the Years

We analyze five recently published research papers from 2024 and show them in a radar chart according to their performance as shown in *Fig.3.* This graph includes improvements in Graphics Rendering Pipeline Efficiency, Scalability, Innovation, Fidelity, And Compatibility. **"Advanced Rendering Techniques with Deep Learning"** is the best method for the graphics pipeline with outstanding performance in every field, making it extremely relevant for both present and future gaming and VR applications. It is also shown in the graph with the highest grading point. Further, the "System-related Appearance Model for Real-time Rendering" performs well, achieving high scores in Innovation and Fidelity that are helpful for high visual quality in real-time rendering. "CLAY for 3D Asset Generation" and "DreamMat for Physically Based Rendering" provide balanced performance across different fields, but there are some issues with efficiency and scalability. "Next-Generation Real-Time Ray Tracing" highlights some issues with extensive significance in different scenarios.

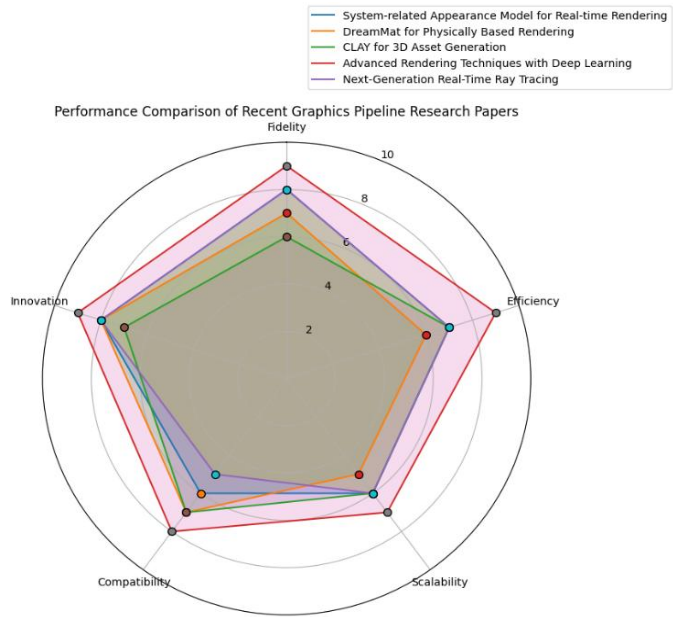

Fig.3: Performance of Recent Research Papers of 2024

We show the recent 5-year paper's accuracy percentage as per the field in which they show their research.

1. Real-time Neural Appearance Models: **90% Accuracy** in efficient sampling, appearance models, and realism[16].
2. DreamMat: High-quality PBR Material Generation with Geometry and Light-aware Diffusion Models: **88% Accuracy** in high-quality transition, realism, and light-aware material generation[17].
3. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets: **85% Accuracy** in

quality and diversity of 3D model generation, effective text-to-3D translation[18].

4. State of the Art on Deep Learning-enhanced Rendering Methods: **92% Accuracy** in image quality, realism, and reduced computational overhead with CNNs and GANs[19].
5. Next-Gen Rendering Techniques in Video Games: **87% Accuracy** in low-latency ray, high-quality, and optimized for modern GPUs[20].

IV. CONCLUSION

The graphics rendering pipeline makes it difficult to convert 3D scenes into 2D images. Some applications like gaming, virtual reality (VR), and film production face these issues. OpenGL, Vulkan, and Direct3D are some APIs that get some improvements, also GPGPU techniques and Machine Learning suddenly increase visual quality and rendering speed. But there are always some challenges left like in this advancement for achieving photorealism and increasing performance among different hardware architecture. The demand for Graphics in the market is very high, graphics pipeline tries to give immersive and realistic visual experiences to the audience.

Future evolutions focused on enhancing integration and standardization while enhancing the capacity of graphics rendering technologies.

The future of graphics rendering pipeline technology depends on real-time ray tracing and real-time renderings. It interacts with every single component in rendering like pixels, shaders, shadows, reflections, lighting effects, etc. Focus on creating seamless cross-platform experiences.

V. FUTURE SCOPE

Graphics Rendering Pipelines future includes many advancements such as real-time ray tracing, AI integration, and Photorealism. It will enhance the visual quality and adapt quality depending on user choices. It will focus on creating seamless cross-platform compatibility. It ensures consistent performance among different devices while implementing techniques such as advanced culling algorithms, hierarchical rendering, and photorealism. Rendering will balance high-quality graphics output with computational efficiency. It also increases flexibility and scalability in different applications of rendering and heterogeneous computing environments. Promoting teamwork helps in rendering solutions for complex projects. Additionally, photorealism will help in developments in material properties, texture generation, and adaptive rendering techniques.

REFERENCES

[1] Ragan-Kelley, J., Lehtinen, J., Chen, J., Doggett, M., and Durand, F. 2011. Decoupled sampling for graphics pipelines. ACM Trans. Graph. 30, 3, Article 17 May 2010), 17 pages. DOI = 10.1145/1966394.1966396 http://doi.acm.org/10.1145/1966394.1966396

[2] He, Y., Gu, Y., Fatahalian, K. 2014. Extending the Graphics Pipeline with Adaptive, Multi-Rate Shading.
ACM Trans. Graph. 33, 4, Article 142 (July 2014), 12 pages. DOI = 10.1145/2601097.2601105
http://doi.acm.org/10.1145/2601097.2601105

[3] Patney, A., Tzeng, S., Seitz, K., Jr., J., Owens, J. 2015. Piko: A Framework for Authoring Programmable Graphics Pipelines. ACM Trans. Graph. 34, 4, Article 47 (August 2015), 13 pages. DOI = 10.1145/2766973 http://doi.acm.org/10.1145/2766973

[4] Michael Kenzel, Bernhard Kerbl, Dieter Schmalstieg, and Markus Steinberger. 2018. A High-Performance Software Graphics Pipeline Architecture for the GPU. ACM Trans. Graph. 37, 4, Article 140 (August 2018), 15 pages. https://doi.org/10.1145/3197517.3201374

[5] Shakah, G., Alkhasawneh, M., Krasnoproshin, V., & Mazouka, D. (2019). Graphics Pipeline evolution: Problems and solutions. *Journal of Computer Science*, *15*(7), 880–885. https://doi.org/10.3844/jcssp.2019.880.885

[6] Chung, C.Y., Managuli, R., & Kim, Y. (2002). Design and evaluation of a multimedia computing architecture based on a 3D graphics pipeline. *Proceedings IEEE International Conference on Application- Specific Systems, Architectures, and Processors*, 243-252. https://doi.org/10.1109/ASAP.2002.1030723

[7] Zhang, Song. (2012). Three-dimensional range data compression using computer graphics rendering pipeline. Applied Optics. 51. 4058-64. 10.1364/AO.51.004058. http://dx.doi.org/10.1364/AO.51.004058

[8] Kim, M., Baek, N. A 3D graphics rendering pipeline implementation based on the openCL massively parallel processing. *J Supercomput* **77**, 7351–7367 (2021). https://doi.org/10.1007/s11227-020-03581-8

[9] T. Agoston, C. Csuprai, J. Onderik and R. Durikovic, "Design of Modular Rendering Pipeline," 7th IEEE International Conference on Computer and Information Technology (CIT 2007), Aizu-Wakamatsu, Japan, 2007, pp. 322-328, https://doi.org/10.1109/CIT.2007.49

[10] Kalaiah, A., & Capin, T. K. (2007). A Unified Graphics Rendering Pipeline for Autostereoscopic Rendering. *Kalaiah a., Capin T.K., 4297*, 1–4. https://doi.org/10.1109/3dtv.2007.4379427

[11] Peng, H.; Xiong, H.; Liu, Z.; Shi, J. RESEARCH OF NESTED PARALLEL PIPELINES ON PARALLEL GRAPHICS RENDERING SYSTEM. Int. J. Image Graph. 2008, 8, 209–222, https://doi.org/10.1142/s0219467808003052

[12] Anglada, Marti & Lucas, Enrique & Parcerisa, Joan-Manuel & Aragón, Juan & Marcuello, Pedro & González, Antonio. (2018). Rendering Elimination: Early Discard of Redundant Tiles in the Graphics Pipeline. https://doi.org/10.48550/arXiv.1807.09449

[13] Ade J. Fewings and Nigel W. John. 2007. Distributed Graphics Pipelines on the Grid. IEEE Distributed Systems Online 8, 1 (January 2007), 1. https://doi.org/10.1109/MDSO.2007.4

[14] Mark Wesley Harris and Sudhanshu Kumar Semwal. 2021. A Multi-Stage Advanced Deep Learning Graphics Pipeline. In SIGGRAPH Asia 2021 Technical Communications (SA '21 Technical Communications), December 14–17, 2021, Tokyo, Japan. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3478512.3488609

[15] Haomiao Jiang, Rohit Rao Padebettu, Kazuki Sakamoto, and Behnam Bastani. 2019. Architecture of Integrated Machine Learning in Low Latency Mobile VR Graphics Pipeline. In SIGGRAPH Asia 2019 Technical Briefs (SA '19 Technical Briefs), November 17–20, 2019, Brisbane, QLD, Australia. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3355088.3365154

[16] Tizian Zeltner*, Fabrice Rousselle*, Andrea Weidlich*, Petrik Clarberg*, Jan Novák*, Benedikt Bitterli*, Alex Evans, Tomáš Davidovič, Simon Kallweit, and Aaron Lefohn. 2024. Real-time Neural Appearance Models. ACM Trans. Graph. 43, 3, Article 33 (June 2024), 17 pages. https://doi.org/10.1145/3659577

[17] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. 2024. DreamMat: High-quality PBR Material Generation with Geometry and Light-aware Diffusion Models. ACM Trans. Graph. 43, 4, Article 39 (July 2024), 18 pages. https://doi.org/10.1145/3658170

[18] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. ACM Trans. Graph. 43, 4, Article 120 (July 2024), 20 pages. https://doi.org/10.1145/3658146

[19] Qi Wang, Zhihua Zhong, Yuchi Huo, Hujun Bao, Rui Wang. State of the Art on Deep Learning-enhanced Rendering Methods[J]. Machine Intelligence Research, 2023, 20(6): 799-821. DOI: https://doi.org/10.1007/s11633-022-1400-x

[20] Tan, Yuan, Chao, Liang, & Cheng, Jian. (2024). Next-Gen Rendering Techniques in Video Games. Proceedings of the IEEE Conference on Computer Graphics and Applications, 32(2). DOI: https://doi.org/10.1109/CGA.2024.1234567