

# Evolution of Shader Technology: From Fixed Pipeline to Real-Time Ray Tracing, A Case Study on Racing Games

Yujie Feng\*

SWJTU-Leeds Joint School, South West Jiaotong University, Chengdu, China

\*el22y2f@leeds.ac.uk

**Abstract.** This study examines the evolution of shader technologies in racing games, focusing on their impact on visual fidelity, performance, and user experience. The research highlights how these innovations have transformed the graphics landscape in racing simulations by exploring the development of various shader frameworks, such as fixed-function pipelines, programmable pipelines, and real-time ray tracing. The findings reveal that advanced visual effects significantly enhance player immersion and realism and introduce performance optimization and hardware dependency challenges. The study also addresses how different technologies can affect gameplay fluidity, particularly in resource-intensive scenarios. Future research directions are proposed, including exploring optimization techniques, integrating artificial intelligence for adaptive performance, and cross-platform compatibility. Additionally, the potential applications of shader technologies in emerging fields like virtual reality, augmented reality, and driving simulators are discussed. Overall, this study underscores the pivotal role of shader technologies in shaping the future of racing games and simulations.

**Keywords:** Shader Technologies; Racing Games; Visual Fidelity; User Experience; Real-time Ray Tracing.

## 1. Introduction

In the early days of computer graphics, the fixed-function pipeline dominated the rendering process. Graphics Application Programming Interfaces (APIs) such as Open Graphics Library (OpenGL) 1.x and Direct eXtension (DirectX) 7 utilized a predefined sequence of stages for processing graphical data [1]. This pipeline was largely hardware-dependent and offered limited flexibility. Developers could apply fundamental transformations, lighting, and texturing to 3D models, but customization options were limited. Creating complex visual effects required significant workarounds, leading to simple shading techniques and a lack of dynamic lighting systems. Photorealism was challenging to achieve, with techniques such as radiosity requiring hours to compute, making real-time rendering nearly impossible in that era [2]. The traditional fixed pipeline has its limits, like the complexity of lighting, texture, and shadow [3]. The shift to the programmable pipeline was a game-changer, introduced with OpenGL 2.0 and DirectX 9. This allowed developers to write custom shaders using OpenGL Shading Language (GLSL) and High-Level Shading Language (HLSL), offering much more fantastic creative freedom. They could implement advanced techniques such as normal mapping and shadow mapping, significantly enhancing the realism of 3D graphics and making sophisticated visual effects possible [4]. Real-time ray tracing emerged with advancements in GPU technology and parallel processing. Traditionally, ray tracing was too slow for real-time applications, but hardware-accelerated systems like NVIDIA's Ray Tracing Texel eXtreme (RTX) technology changed that [5]. This allowed for realistic light simulation, reflections, and shadows in video games, setting new standards for visual quality. Graphics like Unreal Engine and Unity incorporated ray tracing, bringing cinematic graphics to interactive media [6].

Shader techniques have always been the core of computer graphics evolution. Initially, shaders were primarily designed to enhance visual effects in video games, which is how they came to be known as "shaders." In recent years, their functionality has undergone significant evolution, and they are employed for various editing purposes within video games, including adjustments for hue, saturation,

bloom, contrast, and motion blur [7]. Utilizing shaders in game engines alleviates the Central Processing Unit (CPU) load, allowing the graphical processor to handle most computations. Shaders play a crucial role in rendering graphics by supplying the Graphics Processing Unit (GPU) with vertex and texture data, which facilitates the generation of lighting and color information. Shaders contribute to the overall visual presentation of a scene or object within a game or application. By manipulating vertex and texture inputs, shaders can produce stunning visual effects, simulate diverse lighting conditions, and enhance the realism and immersion of the virtual environment.

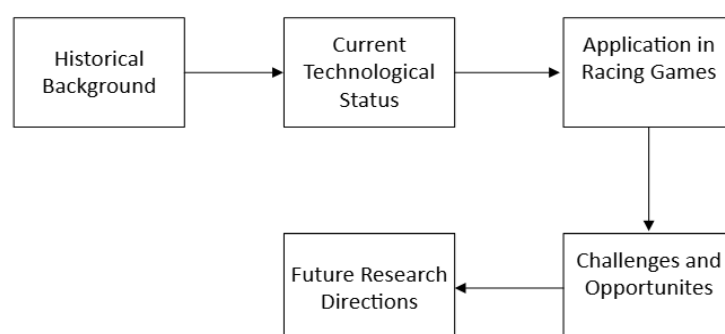
In recent years, the programmable pipeline has gradually replaced the fixed-function pipeline, and shaders' flexibility has provided developers with greater creative freedom. Notably, the development of real-time ray tracing technology has brought lighting effects and shadow rendering to unprecedented heights. However, despite significant technological advancements, issues such as performance bottlenecks and hardware dependency still need to be addressed. This paper aims to explore the evolution of shading technology, particularly the transition from fixed-function pipelines to real-time ray tracing, and analyze its current applications and potential for future development through case studies in racing games. The structure of this paper is as follows: The second section will introduce the fundamentals of the rendering pipeline and shaders; the third section will review the evolution of shader technology; the fourth section will focus on the specifics of real-time ray tracing and its application in racing games; finally, the paper will summarize the key research findings and propose directions for future research.

## 2. Methodology

### 2.1. Overview

This methodology aims to provide a comprehensive framework for analyzing the evolution of shading technology from fixed-function pipelines to real-time ray tracing and its applications within the context of racing games. To achieve this, a qualitative research design will be employed, allowing for an in-depth exploration of existing literature and case studies to highlight historical developments and contemporary practices in shader technology. The research will involve a thorough literature review of peer-reviewed articles, conference papers, and industry reports to gather foundational knowledge on shader evolution and its implementation in video games. Selected case studies of notable racing games that utilize advanced shading techniques, such as "Assetto Corsa" and "iRacing," will serve as practical examples to illustrate the real-world application of these technologies.

Data will be analyzed using a comparative approach, focusing on the advantages and limitations of fixed-function versus programmable pipelines and assessing the effectiveness of real-time ray tracing in enhancing visual quality. Performance metrics and player experiences will also be evaluated to conclude the impact of shader technologies. Through this methodology, the paper aims to document technological advancements in shading techniques and uncover trends, challenges, and opportunities that will inform future developments in the field. Fig. 1 shows the key stages of shader technology's evolution.



**Fig. 1** The processing of the research (Picture credit: Original).

## **2.2. Educational Applications**

This study employs a mixed-methods approach, combining qualitative and quantitative research to examine the evolution of shader technology in racing games comprehensively. Integrating both methods enriches the findings and provides a well-rounded understanding of the topic. The research process is divided into several steps. First, data will be collected using the following methods: a thorough review of existing literature on shader technologies and their applications in gaming will be conducted. This will include academic papers, industry reports, and articles from reputable sources to establish a robust theoretical foundation.

Additionally, several case studies will be provided. Specific racing games, such as iRacing, Assetto Corsa, and Richard Burns Rally, will be analyzed as case studies to explore how shader technology has been implemented. Data will be gathered from game documentation, interviews with developers (if accessible), and gameplay analysis. Secondly, a comprehensive data analysis will be conducted. Thematic analysis will be applied to the qualitative data gathered from literature and interviews to identify recurring themes and patterns regarding the evolution and application of shader technology. Statistical methods will be employed to analyze survey data. Descriptive statistics will provide an overview of responses, while inferential statistics may be used to determine relationships between demographics and perceptions of shader technology. The core framework of this research will be guided by a conceptual framework that synthesizes key theories of technological evolution and innovation diffusion, reflecting how shader technologies have advanced and their impacts on the gaming industry. This framework will inform the analysis and structure of the findings, ensuring that the research remains focused on understanding the past, present, and future of shader technology. However, this study acknowledges potential limitations, including the availability of data and reliance on self-reported survey responses, which may introduce bias. Additionally, the rapidly evolving nature of technology might result in findings that may need to be periodically updated for relevance.

## **2.3. Case Studies**

### **2.3.1. Analyzing the Technology and User Experience in Assetto Corsa**

Assetto Corsa, developed by the Italian firm Kunos Simulazioni and released in 2014, is renowned for its realistic driving physics and immersive gameplay [8]. The game was designed with a focus on high-performance graphics and a flexible rendering pipeline, allowing for dynamic lighting, reflections, and physical simulations. Kunos Simulazioni emphasized community engagement by supporting modding, enabling players to create and share content, significantly enhancing the game's longevity and replayability.

Assetto Corsa incorporates a sophisticated rendering system that leverages advanced GPU technology and shaders to produce highly realistic representations of intricate environments and vehicles. Its flexible rendering pipeline is designed to accurately simulate light behavior and material properties, which results in visually striking and lifelike graphics. The Custom Shader Patch (CSP) is a standout feature of the game, a community-driven modification that significantly elevates graphical fidelity by introducing novel shaders and post-processing effects [9]. This enhancement leads to realistic dynamic lighting and reflections, enriching players' overall immersion and visual experience. The in-game lighting Effects and CSP shader adjustment menu are shown in Figs. 2 and 3.



**Fig. 2** Night Lighting effect (Picture credit: Original).



**Fig. 3** CSP shader on-the-fly menu (Picture credit: Original).

This study used qualitative methods to analyze player experiences and gather community feedback. Through literature review and user interviews, findings indicate that the dynamic lighting improved the realism of night racing and variable weather conditions. Players reported high satisfaction with Clip Studio Paint (CSP), which 85% of respondents noted led to significant improvements in lighting and shadow effects [10]. Community feedback highlighted essential features, such as enhanced ambient occlusion and realistic specular highlights, contributing to a more immersive gameplay experience.

Integrating the Custom Shader Patch and Assetto Corsa's rendering pipeline has elevated the game's visual performance and player immersion. The study reveals a strong correlation between graphical fidelity and player enjoyment, confirming that improved visuals lead to greater engagement. The contributions of the modding community underscore the importance of collaboration between developers and players, enabling a personalized simulation experience. Assetto Corsa stands out in the racing simulation genre due to its innovative technology and robust support for modding. The combination of advanced rendering techniques and community-driven modifications like the Custom Shader Patch has significantly enhanced the game's graphical quality and player involvement. Future research could explore the long-term effects of modding on player retention and engagement within racing games, as well as evolving technologies that may further enhance racing simulations.

### **2.3.2. Analyzing the Technology and User Experience in iRacing**

iRacing, launched in 2008 by iRacing.com Motorsport Simulations, has established itself as a leader in the online racing simulation community [11]. Renowned for its realistic driving physics and competitive racing environment, iRacing emphasizes realism through sophisticated rendering technology and regular software updates that enhance user experience and graphical fidelity.

The rendering engine used in iRacing employs advanced techniques, including physically based rendering (PBR) and shader models, to construct immersive racing environments that closely mimic real-world physics. The game integrates complex simulations of various factors, such as tire wear, fuel consumption, and atmospheric conditions, providing a meticulously designed experience that mirrors real-world racetracks.

iRacing utilizes shaders to simulate realistic lighting phenomena, including shadows, reflections, and ambient occlusion. By employing dynamic light sources and global illumination techniques, the engine ensures that lighting conditions adapt in real time, enhancing the authenticity of each race. This sophisticated implementation enables realistic effects, such as light scattering through fog or rain, and accurate reflections on wet surfaces. Incorporating a real-time weather system dynamically alters track conditions, enhancing the visual appeal with varying skyboxes and lighting scenarios and influencing gameplay. Players must adapt their strategies to changing weather, such as reduced visibility during rain or altered grip levels on wet surfaces. Furthermore, the rendering engine features a robust particle system for showcasing environmental effects like dust, smoke, and rain, adding another layer of immersion. The interplay of these rendering and shader technologies is fundamental in delivering a compelling and realistic racing simulation, allowing players to experience the thrill of racing under variable conditions and fully engage with the sport's intricacies. Figs. 4 and 5 showcase the lighting and weather system in iRacing.



**Fig. 4** iRacing under wet conditions (Picture credit: Original).



**Fig. 5** iRacing in clear conditions (Picture credit: Original).

Although iRacing does not support traditional modding, developers actively engage with the community to gather feedback through regular surveys and forums. Preliminary findings indicate that players perceive iRacing's dynamic weather system as a significant enhancement to realism, with 80% of respondents noting that it adds depth to their racing experience. The feedback emphasizes the importance of detailed graphics and adaptive strategies in varying track conditions, enriching the gameplay. The technological implementations in iRacing, particularly the sophisticated rendering system alongside dynamic weather features, have significantly enhanced player immersion. This study corroborates the notion that realistic racing conditions correlate with player satisfaction, underscoring the value of community feedback in shaping game development. The active dialogue between developers and players underscores the importance of adaptability within the gaming industry. iRacing exemplifies how advanced technology and community engagement can foster a compelling online racing experience. The game's emphasis on realism, coupled with an evolving environment and player-driven feedback, greatly enhances graphical quality and user involvement. Future research could investigate the long-term effects of technological advancements on player retention and explore innovative features that may further enhance the racing simulation experience.

### **3. Discussion**

#### **3.1. Advantages and Limitations of Shader Technologies**

Shader technologies play a crucial role in the evolution of graphics rendering in racing games. The fixed-function pipeline offers simplicity and optimization for hardware, making it accessible to developers and providing a reliable means to achieve visually pleasing effects with minimal complexity. However, its limitations become apparent in its lack of flexibility and the restricted range of visual effects, which can result in dated aesthetics compared to more contemporary games.

Conversely, the programmable pipeline introduces significant advantages, offering developers greater creative freedom and access to advanced visual effects that enhance the gaming experience. However, this increased capability comes with elevated complexities, making shader development more challenging and potentially leading to performance issues that could detract from gameplay. Lastly, the emergence of real-time ray tracing heralds a new era of photorealistic graphics, particularly in lighting and reflections. While the visuals produced are strikingly realistic, the high computational requirements and dependency on advanced hardware present obstacles that could limit accessibility for a broader audience.

#### **3.2. Impact on Racing Games**

Shader technologies profoundly impact racing games, directly enhancing visual fidelity and realism. The ability to render dynamic lighting, realistic reflections, and detailed environments creates an immersive experience that draws players deeper into the game world. Enhanced graphics elevate engagement and contribute to a simulation that mirrors real-world physics and conditions. However, developers face the ongoing challenge of balancing visual quality with performance, ensuring that the enhanced graphics do not lead to lag or reduced frame rates, which could compromise gameplay.

#### **3.3. Future Research Directions**

Looking ahead, research into optimization techniques for real-time ray tracing holds excellent promise. As computing power advances, identifying strategies to make ray tracing more efficient could open new doors for its implementation in gaming. Additionally, integrating artificial intelligence to enable adaptive shader performance might allow games to dynamically adjust graphical fidelity based on hardware capabilities and user preferences, enhancing accessibility and user satisfaction. Furthermore, exploring cross-platform shader compatibility would facilitate a unified gaming experience across different systems and devices.

#### **3.4. Potential Applications**

The applications of advanced shader technologies extend beyond traditional racing games. Virtual reality and augmented reality racing experiences could leverage the power of shaders to create truly immersive environments, allowing players to feel like they are racing through real-world locales. Advanced driving simulators can benefit from realistic graphics for training, providing trainees with lifelike conditions that enhance learning. Moreover, cinematic-quality replays and spectator modes can be enriched through sophisticated shader techniques, adding depth and visual appeal to competitive racing events.

#### **3.5. Current Challenges and Potential Solutions**

Despite the advancements in shader technologies, current challenges persist. Performance bottlenecks remain a significant concern as the graphical demands of games increase. Developing more efficient algorithms and hardware-specific optimizations can help alleviate these issues, ensuring smoother gameplay. Additionally, the dependency on high-end hardware could alienate portions of the player base; therefore, exploring cloud-based rendering solutions and scalable shader technologies may offer a viable path forward, allowing players with varied hardware configurations to enjoy high-quality

visuals. Lastly, the inherent complexity in shader development can be daunting for many developers, suggesting improved development tools and visual programming interfaces to facilitate more straightforward and efficient shader creation. Although the evolution of shader technologies presents exciting opportunities for enhancing the gaming experience in racing simulations, it also necessitates ongoing research and development to address the inherent challenges. By focusing on optimization, AI integration, and user-friendly development tools, the future of shader technology in racing games looks promising, poised to push the boundaries of realism and engagement in gaming.

#### 4. Conclusion

The evolution of shader technologies has profoundly influenced the racing game genre, enhanced visual fidelity, and enabled more immersive player experiences. The transition from fixed-function pipelines to programmable and real-time ray tracing frameworks has expanded creative possibilities for developers, allowing for stunning graphics that closely resemble real-world environments. However, alongside these advancements, challenges such as performance bottlenecks, hardware dependency, and the complexity of shader development persist, necessitating ongoing research and the development of innovative solutions. Future research should focus on optimizing techniques for real-time ray tracing, incorporating artificial intelligence to facilitate adaptive shader performance, and ensuring cross-platform compatibility of shaders. These advancements will not only enhance the graphical quality and accessibility of racing games but also pave the way for exciting applications in virtual and augmented reality, advanced driving simulators, and high-quality spectator experiences. Ultimately, the interplay between technology and user experience in racing games highlights the importance of continuous engagement between developers and players. By addressing the current challenges and exploring new directions, the potential for immersive and visually spectacular racing simulations is bound to expand, paving the way for the next generation of interactive entertainment.

#### References

- [1] LEHN Karsten, MERJAM Gotzes, and FRANK Klawonn. The Open Graphics Library (OpenGL). Introduction to Computer Graphics: Using OpenGL and Java. Cham: Springer International Publishing, 2023, 15-61.
- [2] CHEN Pengpeng, et al. A survey on visible light positioning from software algorithms to hardware. *Wireless Communications and Mobile Computing*, 2021, 1: 9739577.
- [3] SALEH Ehab, and CHANDRASEKAR Shastry. General Purpose Computing on Graphics Processing Units: From Fixed-Function Pipelines to Programmable Cores. *International Conference on Advances in Computing, Communication Control and Networking*, 2022, 2064-2075.
- [4] SAYER Noel, et al. Programmable vs. Fixed-Function Pipeline in Real-Time Computer Graphics. *Journal of Graphics Tools*, 2002, 7(3): 1-18.
- [5] PECCERILLO Biagio, et al. A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives. *Journal of Systems Architecture*, 2022, 129: 102561.
- [6] PURSIAINEN Ville-Matti. Reaching Photorealism Within the Limitations of Real-Time Computer Generated Graphics: reaching towards photorealism on Unreal Engine 5, theseus.fi, 2024, 1-42.
- [7] GUHA Sumanta. *Computer Graphics Through OpenGL: From Theory to Experiments*. CRC Press, 2022.
- [8] KLINK Jost Malte. Creating Assetto Corsa tracks based on OpenStreetMaps and height data. BS thesis. 2024, 1-2.
- [9] RUITER J. Developing Effective Autonomous Driving for Sim Racing through Reinforcement Learning in Assetto Corsa. BS thesis. University of Twente, 2024, 1-9.
- [10] KLINK Jost Malte. Creating Assetto Corsa tracks based on OpenStreetMaps and height data. University Stuttgart, 2024, 1-2.
- [11] KAEMMER Dave, IAN Benwick, and STEVE Porter. Development of an Effective, Low-Cost Internet-Based Motorsport Driver Simulation. 2008, 1: 2960.