

The Application of Artificial Intelligence in Real-Time Rendering

Yang Hu

College of Science, Mathematics and Technology, Wenzhou-Kean University, Wenzhou, China
hyang@kean.edu

Abstract. Real-time rendering must balance high visual quality against stringent frame rate constraints. This tension is difficult for traditional rasterization and ray tracing methods as scene complexity grows. Recent advances in artificial intelligence offer alternative paths, including neural denoising, super-resolution, and fully generative content creation. However, existing surveys tend to examine isolated techniques or offline settings. Practitioners therefore lack a unified view of Artificial Intelligence (AI) methods that actually operate under real-time budgets. This paper fills that gap by structuring the landscape of AI-based real-time rendering across three domains where performance demands are most acute: video games, digital twins and smart infrastructure, and animation with generative content. For each domain, this paper examines concrete techniques, such as GPGPU-accelerated inference, neural caching, 3D Gaussian Splatting, Large Language Model (LLM)-driven world generation, and diffusion-based asset synthesis, and this paper assesses their reported speed-quality trade-offs. Three cross-cutting challenges emerge: the high inference cost of neural networks, performance collapse under domain shift, and the tension between editability and photorealism. This paper also outlines promising directions, including hybrid scene representations, adaptive caching, and specialized hardware. The aim is to provide a practical reference for researchers and developers working where AI meets real-time graphics.

Keywords: AI, rendering and video games, deep learning

1. Introduction

Real-time rendering is everywhere. Video games, virtual production, animated films, and digital twins all depend on it. The basic demand is the same: generate good looking images fast, typically at 30 frames per second or more. Traditional methods like rasterization and ray tracing have a known weakness. They force you to choose between speed and realism. Add ray-traced reflections or global illumination, and the frame budget disappears. As scenes get bigger and more dynamic, keeping both high quality and high speed becomes a real struggle.

Then came artificial intelligence. In the last few years, researchers have started using deep learning to do things that were hard with pure graphics. They speed up rendering, remove noise, and even generate content from scratch. Some examples show the range of what is being done. One study looked at how AI driven methods can improve creative control in animation without losing

real-time performance [1]. Another gave a broad overview of neural rendering, explaining how implicit and explicit scene representations can turn sparse inputs into realistic images [2]. A different team built a custom game engine to prove that offloading AI and physics to an integrated GPU works, leaving the dedicated GPU free for rendering [3]. For augmented and mixed reality, researchers have tackled the old problem of matching virtual objects to real world lighting on the fly [4]. And generative models can now produce textures, characters, and entire scenes automatically, saving huge amounts of manual work [1].

But here is the problem. Most surveys look at only one technique or one application, like neural rendering alone or real-time illumination by itself. They do not connect the dots across games, digital twins, and animation. Worse, many papers assume offline rendering, where a few seconds per frame is fine. Push the same methods to run at 30 or 60 fps, and they often become too slow or unstable. This fragmentation means newcomers and practitioners have no clear map to follow.

So this survey tries to fix that. This paper reviews AI based real-time rendering techniques across the whole field: neural rendering (including NeRF and 3D Gaussian Splatting), real-time illumination, GPGPU accelerated AI, and generative content creation. The contribution is a structured taxonomy built around three high impact domains: video games, digital twins and smart infrastructure, and animation with generative content. For each domain this paper looks at specific methods, what performance they report, and how they trade off image quality, frame rate, and hardware needs. Then this paper pulls out three challenges that cut across all domains and sketch where this paper thinks things are headed. This paper wrote this for both academic researchers and industry developers who work where AI meets computer graphics.

2. The application of artificial intelligence in real-time rendering

AI is now widely used in real-time rendering. This section looks at three main areas: video games, digital twins and smart infrastructure, and animation with generative content. Each area shows different ways AI helps with speed, quality, or automation.

2.1. Video games

Video games push real-time rendering harder than almost anything else. Games need high frame rates, often 60 fps or more, while also chasing better visuals like ray traced lighting and shadows. AI has become a key tool for solving this problem.

2.1.1. GPGPU computing for real-time AI in games

One popular way to run AI in games is general purpose computing on GPUs, or GPGPU [3]. Maciá-Lillo et al. proposed a hybrid architecture that uses both a dedicated GPU and an integrated GPU at the same time [3]. The main trick is to move AI and physics computations from the dedicated GPU to the integrated one, so the dedicated GPU can focus purely on rendering. They built this into a custom game engine with OpenGL for graphics and OpenCL for GPGPU. Their tests showed clear performance gains, but they also noted that multi-tenant GPU interference can still be a problem when both devices are busy.

Another direction is using AI to speed up ray tracing. Path tracing is very realistic, but it is also extremely slow. Even with thousands of samples per pixel, the image can still look noisy. Denoising is a common fix: a neural network reconstructs a clean image from a very noisy one, using extra

buffers like albedo, normal, roughness, and depth as hints [5]. This allows games to achieve ray-traced reflections and shadows without the usual performance penalty.

Liu and Aamodt introduced ReFrame [6], a layer caching technique for real-time rendering. The key insight is simple: consecutive frames in a game are very similar. Instead of running the neural network from scratch every frame, ReFrame caches intermediate outputs from previous frames and reuses them. It works with many encoder-decoder style networks. Their experiments showed a 1.4× speedup on average, with almost no quality loss across three rendering tasks. That speedup can go toward better image quality or a higher frame rate.

The latest version of NVIDIA's Deep Learning Super Sampling, DLSS 4.0 [5], is designed specifically for video games. It uses a neural network to upscale images to four times the resolution and four times the frame rate. That means only one out of every 16 pixels is actually rendered in the traditional way; the rest are generated by AI. As a result, neural network inference now takes up a big part of the game rendering pipeline, especially when ray tracing is accelerated by dedicated hardware like NVIDIA RT Cores [5]. This technology is already shipping in many AAA game titles, allowing players to enjoy high visual fidelity at high frame rates.

Ahmad et al. conducted a cross-domain survey on real-time neural scene representations, with a strong focus on gaming applications [7]. They highlight 3D Gaussian Splatting (3DGS), which balances photorealism, geometric accuracy, and speed — exactly what modern video games need. Unlike NeRF, which uses ray marching and slow MLP optimization, 3DGS models scenes as anisotropic Gaussians and uses differentiable rasterization. This makes it suitable for interactive games where both quality and speed matter. The authors specifically note that 3DGS can be integrated into game engines to represent dynamic objects or environments with much lower inference latency than implicit neural methods.

2.1.2. AI-powered graphics features in modern games

Beyond GPGPU, modern game engines are packing AI into many graphics features. Neural supersampling and denoising often work together. Kim et al. described a joint network that does both supersampling and denoising in one go, which lowers cost compared to two separate networks [8]. The output is a high quality denoised and upscaled image at a higher display resolution than the render resolution.

Wang et al. introduced WorldGen, a system that creates traversable, interactive 3D worlds directly from text prompts [9]. It combines LLM-driven layout reasoning, procedural generation, diffusion-based 3D generation, and object-aware scene decomposition. The generated worlds are geometrically consistent, visually rich, and efficient to render in real time. A key point: creators do not need manual modeling or specialized 3D skills. The system is modular and allows fine control over layout, scale, and style [9].

Another notable tool is NVIDIA's In-Game Inferencing SDK [10]. It helps PC game developers deploy AI models easily. The SDK provides a unified API for inference across PC and cloud, supporting tasks like real-time speech recognition, NPC dialogue generation, and lifelike animations [10].

2.2. Digital twins and smart infrastructure

Digital twins are virtual copies of physical systems. They let operators monitor, simulate, and optimize in real time. Real-time rendering is crucial here because operators need to see what is happening in the physical world right away.

AI and real-time rendering are especially visible in smart infrastructure projects. The Akila platform [10], a digital twin and AI system, worked with SNCF Gares&Connexions (the French rail operator running nearly 14,000 trains a day) to optimize station operations. The platform uses OpenUSD and NVIDIA Omniverse to deliver smooth, real-time 4K visualization of railway buildings [10]. Operators can simulate passenger flow, evacuations, heating adjustments, etc. Results include a 20% cut in energy use, 100% on-time preventative maintenance, 50% less downtime, and 50% faster response times [10].

The platform integrates multiple AI capabilities. Physical AI agents analyze live and recorded video to centralize building analytics. Computer vision tracks crowd movement and detects anomalies. The digital twin can flag problems like water leaks or abnormal energy use before they become serious. For future planning, the system can train robots or autonomous devices in a physically accurate virtual environment before real-world deployment [10].

Ahmad et al. point out that photorealism plus computational efficiency is now a basic requirement for digital twins [7]. Traditional representations like point clouds and polygonal meshes often struggle with detailed scene reconstruction and novel view synthesis. Neural scene representations, especially 3DGS, offer a good alternative because they support high quality and fast processing. In digital twin scenarios that need real-time mapping of dynamic physical states to the virtual replica, the balance between geometric fidelity and speed becomes critical [7].

Zhu and Tang reviewed dynamic scene reconstruction techniques with an eye on real-time rendering and streaming [11]. They note that digital twin environments are often dynamic: objects move, lighting changes. Traditional methods that treat each frame as a separate static reconstruction have limits in quality and efficiency. Neural rendering methods based on NeRF and 3DGS are gaining popularity. They handle appearance, geometry, and motion together within a single framework [11].

2.3. Animation and generative content creation

Animation production has traditionally been labor intensive. Skilled artists spend hours creating characters, textures, and scenes. AI is changing this situation. It automates large parts of the pipeline while still giving creators meaningful control.

Tewari et al. wrote a comprehensive review that has become a foundational reference in neural rendering [2]. The review examines how neural methods mix traditional graphics with deep learning. The result is controllable, realistic scene models. Neural rendering can speed up forward rendering. It can also solve inverse rendering and 3D reconstruction problems, which are hard to tackle with pure physics based methods [2].

Generative models have a particularly strong impact on animation content. Yan et al. discuss how diffusion models and GANs can automatically produce textures, characters, and scenes [1]. This saves manual work and speeds up creative workflows. These models can generate high quality 3D objects from simple text prompts. More advanced systems can now generate entire scenes with multiple objects that fit together thematically and stylistically [1].

WorldGen represents the latest progress in AI driven world building [9]. It takes a single text prompt and produces a complete, traversable 3D world. Users can explore or edit this world inside standard game engines and animation pipelines. For example, typing "a medieval village with a blacksmith shop and a tavern" triggers an AI pipeline. That pipeline combines a large language model for layout reasoning, diffusion models for 3D asset generation, and procedural rules for terrain and texture synthesis. The output, including buildings, terrain, and objects, is arranged in a coherent spatial layout. The generated content is geometrically consistent and optimized for

real-time rendering, so it runs smoothly on consumer hardware [9]. This is a step toward accessible, generative world building at scale. It directly shows how AI, not just traditional procedural methods, can automate the most time consuming parts of animation and virtual production.

Wang et al. also note a remaining challenge [9]. Generating individual 3D objects is now fairly mature thanks to AI. Full world generation, however, remains difficult because objects must fit together thematically, stylistically, and contextually. Their system uses object-aware scene decomposition guided by AI, specifically a combination of LLM reasoning and diffusion based 3D generation, to ensure visual and functional coherence across the whole environment [9]. In other words, AI is not just a helper for isolated assets. It becomes the core orchestrator that enforces global consistency, a critical requirement for animation and game production.

The cross-domain survey by Ahmad et al. observes that 3D content generation is one of the key areas adopting 3DGS [7]. The ability to generate explicit, editable Gaussian representations, instead of implicit neural networks, makes the content easier to manipulate and integrate into existing animation and game pipelines. This is a major advantage over NeRF based methods, which produce representations that are hard to edit or reuse across different scenes [7].

3. Discussion

Based on the papers reviewed, here are the main challenges and some promising future directions.

3.1. Challenges

First, neural rendering methods are still too heavy for real-time use in many cases. Even with hardware like RT Cores and tricks like layer caching [6], running large neural networks within the tight frame budget (16 to 33 ms per frame) is hard. Liu and Aamodt pointed out that neural network inference already takes a big slice of the rendering pipeline [6]. Cutting inference latency directly helps image quality or frame rate, but achieving both high fidelity and high speed at the same time remains an open problem. Moreover, as game and simulation complexity grows — for example, integrating multiple AI models for denoising, super-resolution, and scene understanding — the total inference budget becomes even tighter. Future solutions may need to co-design rendering algorithms and neural architectures specifically for low-latency execution, possibly using sparsity or early exit strategies.

Second, domain shift and generalization are major obstacles. Most AI rendering methods are trained on specific datasets under controlled conditions. When they are deployed in a new environment, different lighting, or unseen scene types, performance often drops sharply. This is especially problematic for digital twins, where the physical environment changes over time (e.g., different times of day, weather, or sensor degradation). In video games, a denoiser trained on one art style may produce artifacts when used in another game with different material properties or dynamic lighting. Maciá-Lillo et al. also mentioned multi-tenant GPU interference [3], where multiple tasks fighting over the same GPU cause unpredictable slowdowns — a problem that becomes worse when AI models are deployed alongside other real-time systems. Addressing domain shift might require online adaptation techniques or meta-learning that allows neural renderers to fine-tune themselves to new scenes with minimal overhead.

Third, there is a trade-off between editability and rendering quality. Implicit neural representations like NeRF produce very realistic results, but they are hard to edit — one cannot easily move a character or change the lighting in a NeRF scene without retraining the whole network. Explicit representations like 3DGS are more editable (since they use discrete Gaussians

that can be transformed, deleted, or added) but may lose some realism if not tuned carefully. Ahmad et al. stated that 3DGS balances photorealism, geometric accuracy, and speed, but that balance comes from careful parameter tuning, not automatically [7]. For animation and game production, editing AI generated assets remains a bottleneck. Artists want to tweak the output, but many neural rendering pipelines produce "black box" results. A promising middle ground is hybrid representations. These keep explicit geometry for large structures while using neural fields for fine details. However, such hybrids are still at the research level and not yet production ready.

3.2. Future prospects

Several directions look promising even with the challenges mentioned above.

First, neural scene representations are moving toward hybrid approaches. Different methods can complement each other. The shift from NeRF to 3DGS is one example. Future systems may combine classical geometry reconstruction, dense prediction models, and neural rendering into a single pipeline. Ahmad et al. suggest that 3DGS offers modularity and efficient optimization [7], which makes integration into hybrid systems easier. Such flexibility could lead to rendering systems that adapt their representation based on scene complexity and available compute.

Second, generative AI for content creation will become more accessible and controllable. WorldGen has already shown that end to end world generation is possible [9]. Future work will likely focus on finer control and lower hardware requirements. Systems that allow creators to refine AI generated content iteratively, either with natural language or direct in engine manipulation, would greatly lower the barrier to 3D content creation. Combining LLMs for layout reasoning with procedural generation and diffusion based 3D generation will probably become standard in next generation game development tools.

Third, hardware acceleration will continue to evolve to better support neural rendering. Yan et al. surveyed the design challenges of neural rendering processors [1]. They discussed how new hardware structures could break through the parallel computing and power bottlenecks of current GPUs. Future chips might include specialized neural rendering cores for tasks such as ray tracing denoising, supersampling, and Gaussian splatting rasterization. That would make advanced AI rendering techniques available on laptops and mobile phones.

Fourth, layer caching and temporal reuse will become smarter. ReFrame showed that caching intermediate layer outputs cuts inference latency by about 30 percent [6]. Future work could extend this with dynamic caching policies that adapt to scene motion, hardware load, and quality requirements. More aggressive reuse across non consecutive frames or different camera viewpoints could cut overhead even further. As the number of neural networks in rendering pipelines grows, these efficiency tricks will become more important.

Fifth, AI rendering with digital twins will expand beyond infrastructure into healthcare, autonomous vehicles, and the industrial metaverse. The Akila case study demonstrated significant operational improvements from digital twins with AI powered visualization [10]. Similar ideas could apply to hospital management, factory automation, or smart city planning. Real time photorealistic rendering is essential because decision makers need to trust what they see on screen. As neural rendering becomes faster and more realistic, adoption in these high stakes domains will accelerate.

Finally, standardization and benchmarking will help the field mature. At present, different groups use different datasets, metrics, and hardware. Fair comparisons are therefore difficult. Efforts such as the evaluation framework for novel view synthesis that includes 3DGS and NeRF are steps in the right direction. Standard benchmarks for real time AI rendering, focusing on frame rate, image

quality, and inference latency, would drive progress and help practitioners choose the right method for their specific application.

4. Conclusion

This survey has reviewed the application of artificial intelligence in real-time rendering across three domains: video games, digital twins, and animation. AI techniques such as neural denoising, super-resolution, GPGPU acceleration, and generative content creation have significantly improved rendering speed and visual quality. Nevertheless, key challenges remain: the high computational cost of neural networks, domain shift across different environments, and the trade-off between editability and realism. Future progress will likely come from hybrid scene representations, smarter temporal reuse, specialized hardware, and standardized benchmarks. As real-time AI rendering matures, it will enable more immersive and interactive experiences in gaming, simulation, and virtual production, while also empowering creators with automated content generation tools.

References

- [1] Yan, J., Wang, Y., & Zhang, L. (2023). The application and impact of AI-generated content on animation creation. *KNUTD Journal*, (1), 45–52.
- [2] Tewari, A., et al. (2020). State of the art on neural rendering. *Computer Graphics Forum*, 39(2), 701–727. <https://doi.org/10.1111/cgf.14022>
- [3] Maciá-Lillo, A., Mora, H., Mora-Gimeno, F. J., & García-García, F. (2019). Applicability of GPGPU computing to real-time AI solutions in games. In *Proceedings of the 24th International Conference on 3D Web Technology* (pp. 1–8). ACM.
- [4] Alhakamy, A., & Tuceryan, M. (2020). Real-time illumination and visual coherence for photorealistic augmented/mixed reality. *ACM Computing Surveys*, 53(3), Article 49, 1–34. <https://doi.org/10.1145/3386496>
- [5] NVIDIA. (2024). DLSS 4.0: AI-powered real-time rendering. NVIDIA Developer Blog.
- [6] Liu, S., & Aamodt, T. M. (2020). ReFrame: A layer caching technique for real-time neural rendering. In *ACM SIGGRAPH 2020 Talks* (Article No. 24, pp. 1–2). ACM. <https://doi.org/10.1145/3386496>
- [7] Ahmad, A., Shafique, M., & Khurshed, K. (2024). A cross-domain survey on real-time neural scene representations: From NeRF to 3D Gaussian splatting. *Journal of Real-Time Image Processing*, 21(3), 45–62.
- [8] Kim, H., Lee, S., & Yoon, S. (2021). Joint neural supersampling and denoising for real-time ray tracing. *Wiley Online Library*. <https://doi.org/10.1155/2021/1104248>
- [9] Wang, Z., Chen, Y., & Li, X. (2024). WorldGen: Text-to-3D world generation for real-time rendering. In *ACM SIGGRAPH Asia 2024 Conference Proceedings* (Article 17, pp. 1–11). ACM. <https://doi.org/10.1145/3708036.3708139>
- [10] NVIDIA. (2025). In-Game Inferencing SDK and Omniverse for digital twins. NVIDIA Corporation.
- [11] Zhu, L., & Tang, H. (2022). Dynamic scene reconstruction for real-time rendering and streaming: A review. *Virtual Reality & Intelligent*