DOI: 10.54254/2755-2721/71/20241669

Deep learning-driven optimization of real-time ray tracing for enhanced immersive virtual reality

Qiongxian Zhang[1,†], Ruonan Shi[2,†], Miao Geng[3,4,*]

[1]Monash University, Clayton, Australia
[2]University of Melbourne, Parkville VIC 3010, Australia
[3]University of London, London, United Kingdom

[4]rara481846778@gmail.com
*corresponding author
[†]Qiongxian Zhang and Ruonan Shi contributed equally to this work.

Abstract. Photorealistic rendering is essential for immersive virtual reality and real-time ray tracing is required for this, yet the necessary computational load has so far hindered its use in scenarios where a fast response and high frame rates are paramount. To overcome this challenge, we present a deep learning-based model that optimises the computational load of ray tracing through utilising convolutional neural networks (CNNs) to predict light—scene interactions. We achieve this by training CNNs from datasets with offline ray-traced images. Using this learning process, we approximate the contents of light transport simulations from scene point sampling. The use of this optimised algorithm in virtual reality scenes, thus, enables renderings of more complex scenes, with dynamic lighting, complex geometry and interactive elements under high frame rates, keeping users immersed and responsive to events. Our results show that this method achieves visual similarities to traditional ray tracing, which is photorealistic rendering, but can be performed in real-time in virtual reality. We anticipate this work leads to many potential applications of ray tracing in many scenarios in virtual reality, such as gaming, architectural rendering and training.

Keywords: Real-time ray tracing, deep learning, convolutional neural networks, virtual reality, immersive environments.

1. Introduction

Many recent breakthroughs in computer graphics research have been motivated by the quest to achieve increasingly realistic visual rendering, especially in the case of virtual reality (VR). The development of ray tracing as a technique is one such major advance, and it allows for a realistic simulation of light-matter interactions, such as reflection and refraction. However, to render a complex scene at high quality using ray tracing, it is often necessary to trace the path of one or more rays for each pixel in the output image. As the number of pixels in today's VR scenes can easily exceed 1 million, ray tracing quickly becomes computationally prohibitive. Traditional ray tracing algorithms also pose a number of technical challenges in a real-time VR setting, such as performing collisions tests between traced rays and all objects in a scene. Additionally, existing techniques typically provide correct solutions only for relatively simple materials, without the ability to simulate more intricate effects, such as translucency or volumetric scattering. Despite these drawbacks, existing ray tracing techniques are known to yield high-fidelity rendering results, and computer graphics researchers have hence been pursuing alternative methods that retain the fidelity of ray tracing while reducing the computational burden [1]. One intriguing possibility is to exploit deep learning, and in particular, convolutional neural networks (CNNs). Due to their architecture and training procedure, CNNs can be highly effective at recognising patterns and can, in many cases, generalise any results sufficiently to be applicable to unseen inputs.

2. Optimization of Ray Tracing Algorithms Using Deep Learning

2.1. Convolutional Neural Networks for Ray Tracing

Convolutional neural networks (CNNs) have been used to great advantage in many areas of computer vision, for which they are blazingly effective – and also showing promise as an efficient way to reduce the computational cost of render passes. In the context of ray tracing, CNNs can try to guess what light will do based on patterns learned from very large sets of ray-traced images. In other words, the CNN train its own model of light transport on a large corpus of ray-traced data, and use that approximation rather than perform the computationally costly light transport simulation. It can reliably guess how the maths of indirect lighting will play out, massively reducing the computational burden in the render pipeline. One of the great things about this approach is that the CNN is able to generalise away from the training data (that is, to guess what will happen in unseen combinations of scenes and materials). This means that, in general, it can provide real-time ray traced virtual reality rerenders at frame rates where a more direct approach would struggle, even for super-complex, ever-changing VR scenes. In particular, the ability to move on from static datasets to dynamic ones should mean that light and material variants can be calculated in real time, as the viewer moves within the world. This is crucial for interactive VR, where a scene can transform dynamically based on the user changing the lighting source or moving the objects [2]. In this case, dynamic changes on light sources or geometry position would just be recalculated by the CNN without missing a beat. Better still, a CNN model can be trained, then tweaked using techniques known as transfer learning, to better handle particular types of scenes or objects.

We can model the CNN operation of a ray-tracer mathematically by zeroing in on the convolution operation that lies at the heart of how the network processes input images and makes predictions about light interactions. The convolution operation on the CNN can be expressed as:

$$Z_{i,j,k}^{(l)} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{c=1}^{C^{(l-1)}} W_{m,n,c,k}^{(l)} \cdot X_{i+m-1,j+n-1,c}^{(l-1)} + b_{k}^{(l)} \tag{1}$$

This formula illustrates how the CNN computes each element of the feature map in layer l by convolving the input from the previous layer with a set of filters $W^{(l)}$ and adding a bias term $b^{(l)}$.

2.2. Integrating Deep Learning with Traditional Ray Tracing

This combination of DL with classic ray tracing requires a careful balancing act between what DL is good at (pattern recognition and approximation) and what classic ray tracing is good at (accuracy and all necessary details). The hybrid model can work by having CNNs handle primarily the initial stages of the light interaction calculations – such as global illumination, shadow mapping and so on – while the classic ray tracing algorithm takes care of refining these predictions, and of individual cases (eg, caustics and reflections) where the additional accuracy is required. The advantage of this model is that it substantially reduces the number of rays that need to be traced (hence, the computational load), and is especially important to immersive VR applications that may not possess a brute-force ray tracer as an option. The secret behind this combination is a smooth handoff between the CNN and the classic ray-tracing engine: the CNN quickly determines the lighting conditions of the general scene and provides a rough approximation of it, while the classic ray-tracer refines it – adding detail and accuracy where it can and where it is needed – while spending additional computational load only in areas where the CNN's approximations are not good enough (eg, where caustics occur, or when many different light bounces have to be accounted for, as in typical real-world light scattering scenarios) [3]. A more advanced integration can use adaptive sampling – one can dynamically adjust the number of rays traced by the classic ray-tracer based on the confidence levels of the CNN's predictions.

2.3. Training and Implementation Considerations

To train the CNN to predict light interactions accurately in the ray-traced scene, we used a dataset of pre-rendered scenes with varying complexity, annotated with fine-grained light transport information. During the training, the network was guided to learn expensive light interaction information by gradient descent with backpropagation. Once trained, the optimised CNN was implemented in the GPU-accelerated rendering pipeline, enabled by real-time inference during ray tracing. The implementation of the whole pipeline also involves careful tuning of the trade-off between speed and accuracy, especially for high-frequency features such as fine textures and sharp edges under different visual conditions, while the dataset selection ensures that the CNN can handle the wide range of visual conditions in VR environments, including not only soft shadows and subtle gradients but also sharp highlights and dramatic specular reflection. During the training, data augmentation and network regularisation are employed to mitigate the overfitting issue, while iterative test and validation are also performed to tune the inference performance by comparing the CNN's prediction against the ground truth ray-traced images [4]. The GPU implementation further takes advantage of the parallel processing capabilities, which reduces the inference time significantly and enables real-time application. The trained CNN is then integrated into the hybrid rendering pipeline, able to quickly and accurately predict the light interaction, offering a solid foundation for the entire pipeline [5].

3. Application in Immersive Virtual Reality

3.1. Enhancing User Immersion Through Optimized Rendering

Given the importance of visuals for rendering a virtual environment, the realism of visual rendering is a crucial factor for user immersion for VR. In contrast to legacy rendering techniques, optimised ray tracing algorithms with deep learning provide a hyper-realistic rendering of scenes, which can be rendered at the requisite frame rates for VR and thereby maximise the user experience. Moreover, by increasing the computational efficiency of ray tracing, our approach reduces the computational burden typically associated with more real-looking, detailed scenes with more dynamic natural light and a very high complexity of geometric objects that are important to create natural-looking environments and hence believable virtual worlds. The increased level of realism also improves immersion, which in turn enables more natural user interactions within that environment (based on fidelity of the rendered image matching natural physical behaviour of such visuals). For instance, an improved user experience in a VR architecture walkthrough application could allow the user to navigate across different rooms within that building showcased in the VR environment. The position and intensity of light sources across the space change in real time as the user moves from one room to another. The shadows cast, reflections on different surfaces and diffused light all dynamically adapt to changing light conditions that ultimately offer the user an experience that resembles natural light behaviour. This level of realism is crucial for achieving a sense of presence in the virtual environment, the well-known key to achieving immersive VR [6]. Furthermore, the ability to render complex scenes at high frame rates, especially when the scene is changing (such as the user moving around in that VR environment), is crucial to avoiding the risk of motion sickness and achieving the best possible user experience. Table 1 illustrates the impact of optimised ray tracing on VR immersion, comparing the rendering features before and after optimisation [7].

Table 1. Impact of Ray Tracing Optimization on VR Immersion

Rendering Feature	Before Optimization (ms)	After Optimization (ms)	Impact on Immersion
Dynamic Lighting	150	80	Increased realism in various lighting conditions.
Real-Time Shadow Adjustments	120	60	Smooth transitions and adjustments in shadows.
Complex Geometry Rendering	180	90	More detailed and accurate representation of objects.
Reflection and Refraction	200	100	Improved visual fidelity in reflective and refractive surfaces.
High Frame Rate Maintenance	160	70	Seamless experience with minimal latency.

3.2. Real-Time Interaction and Responsiveness

One of the most important constraints for VR applications is responsiveness – any noticeable lag or delay between the interacting user and the rendering system can be very disruptive and can even induce motion sickness. The optimised ray tracing algorithm continues to enable real-time interaction in scenes with complex lighting and geometry, and a balance between computational cost and rendering fidelity is crucial. The CNN offloads some computational work so that the system can dedicate more computation to user input and scene updates in real-time. For example, in many high-end VR applications, there is a need to keep up with the frame rates when there are rapid movements and quick reactive times [8]. The optimised algorithm allows us to instantly change the lighting and shading based on the user movement, such that the rendered output at any time can keep up with the user action. This responsiveness is not only important in increasing the sense of realism in the game but also to ensure the user immersion without any disruption. For instance, when you pick up a virtual cup, it should appear to be the cup that you are moving around in real time instead of providing any angled distraction that can interrupt your gaming experience [9]. In addition, the decreased computational load also allows for a wider range of complex interactions – such as real-time physics motions (eg, cloth-simulation applications), analysis of more complex geometries, or dynamic environment changes.

3.3. Application to Various VR Scenarios

The optimised ray tracing algorithm could be applied to various VR scenarios from games and entertainment to educational simulations and architectural visualisations. In a game, the real-time rendering of a dynamic environment such as an indoor or outdoor scene with the accurate lighting from various light sources creates a more immersive and impactful narrative and emotional engagement of the player. With the accurate representation of light and shadow, the realism and engagement of the simulation in an educational scenario or learning game improves. In that context, the realistic rendering of a virtual lab or historical reconstruction promotes learning effectiveness. Likewise, in an architectural visualisation, the optimised ray tracing algorithm allows for precise representations of different materials and lighting conditions, which is necessary for architects and designers to give clients a clear understanding of what they will be seeing during day or night and at differing seasons of the year. The algorithm provides a realistic preview of how light interacts with materials of various colours, textures and diffusivities. Such a preview facilitates experimenting with different lighting configurations and materials within an interactive virtual environment to ultimately produce a design that meets the client's expectations and aligns with their vision [10]. In VR-based training simulations, such as those used for medical or military training, the high-fidelity rendering of the illuminant spectrum could further improve the realism of the training environment and training outcomes. The versatility of this algorithm shows that it can be applied to a wide variety of VR projects, from games and entertainment to educational simulations and architectural visualisations.

4. Discussion

4.1. Comparison with Traditional Ray Tracing Methods

However, traditional ray tracing is too slow for such real-time applications, especially in VR where a high frame rate is crucial. The overhead is due to the fact that rays must be traced for every part of the scene. Our optimisation approach significantly reduces this overhead by limiting the number of rays that need to be traced. This leads to significant improvements in computational and memory efficiency and permits real-time ray tracing, including for VR. The efficiency gain has the potential to revolutionise VR by enabling much more complex and dynamic VR environments. For example, in a traditional ray tracing pipeline, one could think about rendering a complex scene with multiple light sources and complex global illumination. Tracing the full set of rays is computationally demanding and leads to very high rendering times which are not suitable for real-time applications. Our optimisation approach leverages the predictive capabilities of CNNs to estimate the global illumination and handle less complex light interactions, dramatically reducing the overhead of ray tracing. Along with an acceleration of the rendering process, our method frees up computational resources that can be leveraged to add more elements in the scenes such as dynamic objects, complex materials and interactivity [11]. Figure 1 compares the computational power required for different rendering tasks using traditional ray tracing methods versus the optimized ray tracing approach.

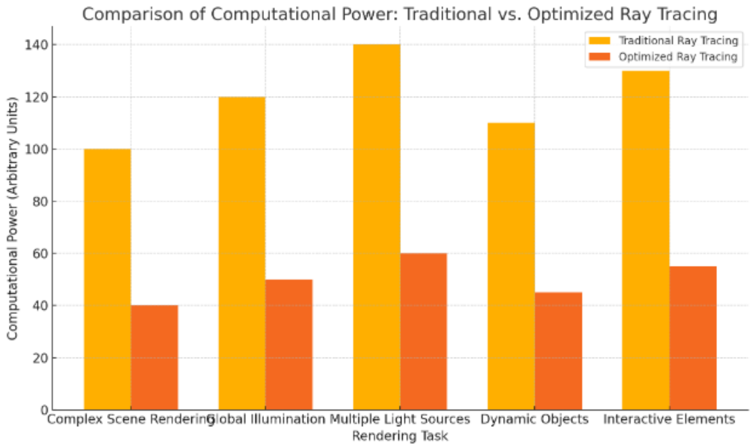

Figure 1. Comparison of Computational Power

4.2. Limitations and Challenges

Despite these advantages, this hybrid approach still poses challenges when optimising a CNN to accurately predict lighting conditions in highly complex scenes that contain unusual geometries or material properties that are not well-represented in the training data. Using traditional ray tracing with the CNN just to clean up the results will still leave the potential for artifacts or inaccuracies in the final result. For example, if the CNN is unable to predict the lighting interactions correctly with very reflective or refractive surfaces, such as glass or water, these scenes can be easily spotted as being less realistic than the rest. Furthermore, the training process itself is costly, and requires access to high-quality training data and a great amount of energy to train. This is not a feasible option for small development teams or projects with limited resources, as the initial set-up and training may require significant time and investment. Another challenge is integrating the CNN into the existing rendering pipeline, which may require significant modifications to the workflow. This may include reworking existing shaders, rewriting the rendering pipeline to accept the CNN's output, and also making sure that the output is compatible with a wide range of VR hardware and software. These are all challenges that need addressing in the future with further research and development of this technique, to guarantee that the optimisation method is as robust as possible and can easily be adapted to a variety of VR environments and applications [12].

5. Conclusion

This work demonstrates the potential of deep learning to drive real-time ray tracing, in particular for immersive virtual reality. Our hybrid ray tracing technique is able to leverage convolutional neural networks embedded into the pipeline, greatly reducing the computational cost of rendering challenging scenes. In contrast to conventional ray tracing, the CNN algorithm supports real-time modification of lighting and geometry, while maintaining the high frame rates necessary for an immersive VR experience. The visual quality is close to that produced by traditional ray tracing but the lower computational load supports more complex scenes and interactions, enhancing the immersion of the user experience. Challenges remain, including generalising the CNN model to support a broader range of visual scenes, as well as integrating the optimised pipeline into a VR system. Future research will focus on addressing these challenges and exploring refinements to the deep learning models to support additional applications of this technology in other areas of computer graphics. This research contributes to the current effort to make photorealistic rendering feasible in real-time applications, and in particular in the nascent field of virtual reality.

References

[1] Yapar, Çağkan, et al. "Real-time outdoor localization using radio maps: A deep learning approach." IEEE Transactions on Wireless Communications 22.12 (2023): 9703-9717.

[2] Zeltner*, Tizian, et al. "Real-time neural appearance models." ACM Transactions on Graphics 43.3 (2024): 1-17.

[3] Hoydis, Jakob, et al. "Sionna RT: Differentiable ray tracing for radio propagation modeling." 2023 IEEE Globecom Workshops (GC Wkshps). IEEE, 2023.

[4] van Wezel, Chris S., et al. "Virtual ray tracer 2.0." Computers & Graphics 111 (2023): 89-102.

[5] Mokayed, Hamam, et al. "Real-time human detection and counting system using deep learning computer vision techniques." Artificial Intelligence and Applications. Vol. 1. No. 4. 2023.

[6] Norouzi, Armin, et al. "Integrating machine learning and model predictive control for automotive applications: A review and future directions." Engineering Applications of Artificial Intelligence 120 (2023): 105878.

[7] Tran, Nghi C., et al. "Anti-aliasing convolution neural network of finger vein recognition for virtual reality (VR) human–robot equipment of metaverse." The Journal of Supercomputing 79.3 (2023): 2767-2782.

[8] Saravanan, G., et al. "Convolutional Neural Networks-based Real-time Gaze Analysis with IoT Integration in User Experience Design." 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS). IEEE, 2023.

[9] Dallel, Mejdi, et al. "Digital twin of an industrial workstation: A novel method of an auto-labeled data generator using virtual reality for human action recognition in the context of human–robot collaboration." Engineering applications of artificial intelligence 118 (2023): 105655.

[10] Hernández, Quercus, et al. "Thermodynamics-informed neural networks for physically realistic mixed reality." Computer Methods in Applied Mechanics and Engineering 407 (2023): 115912.

[11] Maskeliūnas, Rytis, et al. "BiomacVR: A virtual reality-based system for precise human posture and motion analysis in rehabilitation exercises using depth sensors." Electronics 12.2 (2023): 339.

[12] Radanovic, Marko, Kourosh Khoshelham, and Clive Fraser. "Aligning the real and the virtual world: Mixed reality localisation using learning-based 3D–3D model registration." Advanced Engineering Informatics 56 (2023): 101960.