Optimización de la Arquitectura de Computadoras Impulsada por Inteligencia Artificial: Un Enfoque en el Rendimiento de los Videojuegos

Julián David Pérez Uribe, Jose David Silva Solano, Sebastián Mantilla Serrano
Escuela de ingeniería de Sistemas, Universidad Industrial de Santander

***Resumen*—La integración de la inteligencia artificial en las arquitecturas de computadoras ha revolucionado el rendimiento de los videojuegos. Este artículo examina cómo diversas tecnologías han transformado significativamente el renderizado y la eficiencia gráfica. Presentando así, a la Nvidia RTX, que ha emergido como pionero al acelerar el trazado de rayos mediante el uso optimizado de GPU, ofreciendo mejoras notables en la calidad visual y el realismo de las escenas virtuales. Luego, a TensorRT, que optimiza el procesamiento de redes neuronales en las GPU de Nvidia, mejorando considerablemente la eficiencia de inferencia y permitiendo aplicaciones más rápidas y precisas de inteligencia artificial en entornos de juego. Por último, a la tecnología de DLSS (Deep Learning Super Sampling), que introduce técnicas avanzadas de superresolución, permitiendo así elevar la resolución de imágenes renderizadas, como 1080p o 1440p, a niveles más altos como 4K u 8K, sin comprometer significativamente la calidad visual. Esta innovación no solo optimiza el rendimiento del juego al reducir la carga sobre la GPU, sino que también mejora la experiencia visual al mitigar problemas como el aliasing y mejorar la claridad de los detalles visuales.**

Este estudio profundiza en los avances tecnológicos mencionados y explora sus implicaciones futuras para la evolución continua de la experiencia de videojuego. Se destacan las posibilidades de futuras mejoras en el rendimiento, la calidad visual y la eficiencia de las tecnologías existentes, así como el desarrollo de nuevas herramientas y algoritmos que podrían redefinir aún más los estándares de calidad en los videojuegos.

***Palabras clave*—Renderizado, fotorealismo, aceleración de hardware, GPU, kernel, aprendizaje profundo, aliasing, vram.**

I. Introducción

En la última década la inteligencia artificial, la cual hace referencia a la creación de máquinas y sistemas informáticos que simulen la inteligencia humana [1], ha sido una temática particular que ha surgido debido a la gran aplicabilidad y potencial que tiene en diversas áreas del conocimiento y de la vida cotidiana para solucionar problemas o automatizar procesos.

El avance que se observa hoy en día, ha sido el resultado del trabajo de muchos años, desde los planteamientos de Alan Turing, donde formulaba que una máquina podría simular un comportamiento inteligente como el de un ser humano, hasta las reglas de simulación del pensamiento humano y los intentos iniciales de *chatbots*. Aunque estos primeros esfuerzos no tuvieron el éxito esperado en su momento, sentaron las bases para los desarrollos actuales.

Gracias al crecimiento acelerado de la tecnología, donde cada año van surgiendo mejores componentes con mayores capacidades de cálculo que permiten explotar en gran medida el potencial de la IA, en este contexto, se busca aprovechar los beneficios de la IA para mejorar la generación de imágenes y el rendimiento de los videojuegos mediante el uso de GPU's. Este trabajo explora cómo la optimización de la arquitectura de computadoras impulsada por inteligencia artificial puede mejorar el rendimiento en el ámbito de los videojuegos, destacando tecnologías clave como Nvidia RTX, TensorRT y DLSS.

II. Metodología de Investigación

Para el desarrollo del presente artículo, se trazó como meta inicial explorar el impacto de diferentes tecnologías de inteligencia artificial en la optimización de la arquitectura de computadores, con un enfoque particular en el rendimiento de videojuegos. Una vez planteado el objetivo anterior, se ha realizado la búsqueda de artículos científicos relacionados con las tecnologías Nvidia DLSS, RTX y TensorRT, así como estudios sobre la optimización de programas informáticos utilizando IA, con el fin de identificar áreas de interés y metodologías previamente usadas en otras investigaciones.

Seguidamente, se presentan observaciones en busca de evaluar estas tecnologías en algunos escenarios, apoyándose en el hardware de Nvidia que integren las tecnologías mencionadas.

III. Resultados

La integración de la inteligencia artificial (IA) en la optimización de la arquitectura de computadoras ha expandido de forma relevante los límites del rendimiento computacional, en especial en aplicaciones de alto rendimiento como los videojuegos. La destreza de la IA para analizar y optimizar procesos complejos ha demostrado ser invaluable, y un ejemplo sobresaliente es la tecnología de Nvidia. Esta tecnología permite a la IA, desde su propia arquitectura de hardware, analizar y optimizar procesos complejos, utilizando la aceleración de hardware del trazado de rayos para mejorar la calidad visual y el rendimiento en tiempo real de los juegos de video.

III-A. Evaluación del Impacto de la Nvidia TensorRT en la Mejora del Rendimiento Gráfico

Aparte de destacar el gran aporte de la tecnología Nvidia RTX al rendimiento gráfico, es importante mencionar el aporte

de Tensor RT, otra herramienta de la misma compañía, la cual es una plataforma útil para acelerar el rendimiento de aplicaciones de procesamiento de imágenes [1], lo cual permite que los desarrolladores involucrados en esta área puedan crear aplicaciones de procesamiento de imágenes más eficientes y rápidas.

Para describir esta tecnología de una manera más clara, TensorRT es una biblioteca que optimiza y ejecuta redes neuronales entrenadas en GPU's Nvidia [2] que busca maximizar la eficiencia de la inferencia de modelos de inteligencia artificial, que destaca por las siguientes características:

- La optimización de modelos de aprendizaje profundo para su ejecución en GPU's mediante algunas técnicas, como la cuantificación, la fusión de capas y la eliminación de operaciones redundantes, lo que mejora la velocidad de ejecución y la reducción del tamaño del modelo.
- La mejora del rendimiento de la inferencia, donde Tensor RT permite la inferencia de precisión mixta, lo que facilita el uso de diversas precisiones (FP32, FP16, INT8) para mejorar la eficiencia sin sacrificar significativamente la precisión del modelo.
- La fusión de capas de un modelo en una sola operación optimizada que ayuda a reducir el sobrecoste (*overhead*) de memoria y acelerar el procesamiento.
- Un planificador de ejecución que optimiza la asignación de recursos y la secuencia de operaciones en tiempo de ejecución, logrando un dinamismo según la carga de trabajo.
- Una integración con otras herramientas de aprendizaje profundo como lo son TensorFlow, PyTorch y ONNX, facilitando la importación y exportación de modelos de IA desarrollados en estos entornos.

Por ejemplo, TensorRT acelera uno de los modelos de IA generativa más populares, como lo son Stable Diffusion y SDXL, que son modelos que generan imágenes digitales de alta calidad a partir de descripciones de texto, acelerándolo en un 40 % [3]. A continuación, un ejemplo gráfico de esta mejoría.

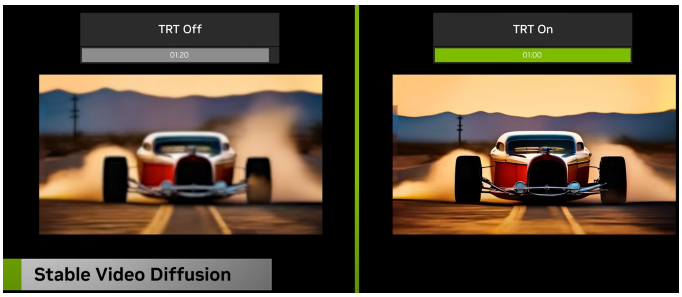

Figura 1. Captura del vídeo publicado por Nvidia, donde demuestra que TensorRT activado genera un vídeo más rápido en comparación al no uso de esta herramienta [3]

El TensorRT, aunque inicialmente diseñado para la optimización de modelos de IA, juega un papel fundamental en mejorar el rendimiento general del sistema, incluido el procesamiento de imágenes en tiempo real, lo cual es vital para los videojuegos modernos. Al integrarlo con otras tecnologías como RTX y DLSS, se logra una sinergia que no solo mejora la eficiencia del procesamiento de imágenes, sino que también contribuye al rendimiento fluido y de alta fidelidad visual en los videojuegos.

III-B. Análisis del Desempeño de la Nvidia RTX

El trazado de rayos es la piedra angular del fotorrealismo [4], debido a que es un método de renderización utilizado en las tarjetas gráficas de las computadoras para determinar el color de los píxeles que componen las imágenes que se muestran en la pantalla, o que se crea en el disco duro mientras se realiza la renderización. Este método funciona lanzando un rayo desde el punto de vista de la cámara, el cual traza su trayectoria mientras interactúa con los objetos de la escena hasta llegar a una fuente de luz, recopilando y almacenando información sobre el color de los objetos con los que interactúa. Al imitar el comportamiento físico de la luz, ofrece resultados de mayor calidad y fotorrealismo que otros métodos utilizados, como sombras suaves y detalladas, oclusión ambiental, refracciones y reflejos precisos. Sin embargo, estas ventajas tienen un costo significativo: la velocidad de renderizado [5]. Es por ello que se utiliza hardware avanzado, así como algoritmos directamente en la CPU de las computadoras para realizar la aceleración de hardware y realizar funciones específicas más rápidas.

Las primeras soluciones de hardware dedicadas al trazado de rayos fueron las tarjetas PCI, si bien solo rastreaban rayos primarios, ya implementaban técnicas para aumentar la eficiencia del rastreo paralelo, como agrupar rayos para aprovechar la coherencia del acceso a la memoria [4]. Más adelante, con la tecnología del chip SaarCOR y un nuevo algoritmo de trazado de rayos, se logró mitigar la latencia del acceso a la memoria al atravesar simultáneamente un grupo de rayos, cargar datos para el siguiente y realizar la intersección en otro grupo [4].

Sin embargo, estas tecnologías eran fijas y no implementaban la aceleración por hardware actual. Por ello, surgió la RPU (Unidad de Procesamiento de Rayos), que admitía sombreadores personalizados con funciones como llamadas a funciones recursivas, instrucciones de seguimiento para iniciar el seguimiento de un rayo arbitrario e instrucciones de carga asíncrona para ocultar la latencia de la memoria [4]. La tecnología de aceleración por hardware ha seguido evolucionando, y hoy en día existen múltiples marcas y productos que la ofrecen para optimizar el renderizado de videojuegos, como Nvidia.

Para enfatizar el rendimiento de las tarjetas gráficas Nvidia RTX y la tecnología que utilizan, en [4] se realizaron experimentos con algoritmos de seguimiento de rutas en la RTX y se compararon con la implementación de código abierto de seguimiento de rutas en Hydra Renderer. Las conclusiones fueron:

1. La Nvidia RTX está dirigida sobre todo a acelerar el acceso aleatorio a la memoria durante el rayo rastreo. Esto se observa en la Figura2
2. Se implementa una agrupación/clasificación de fotogramas por segundo, lo que se confirma por el hecho de que en escenas simples la implementación de hardware no

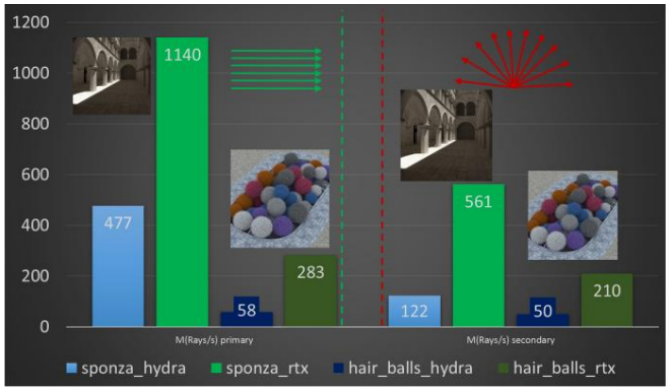

Figura 2. Hydra vs. Nvidia RTX [4]. El sector izquierdo (verde) muestra el rendimiento de los rayos primarios (coherentes) y el derecho (rojo) el rendimiento de los rayos secundarios (aleatorios).

tiene una caída significativa en el rendimiento al pasar de rayos primarios a secundarios. La implementación del software, por otro lado, experimenta una degradación mucho más rápida, posiblemente debido a la creación de trabajo de GPU.

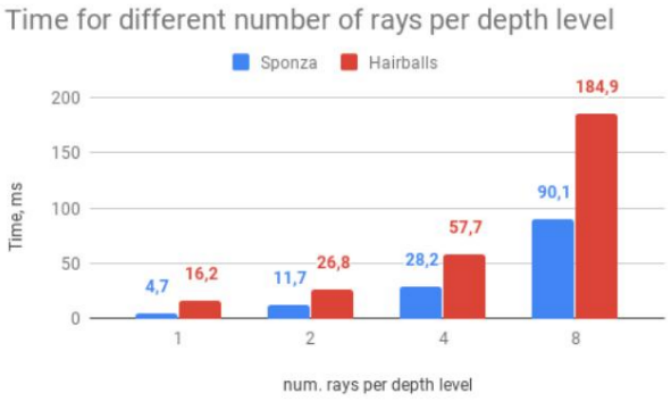

Figura 3. Tiempo empleado por trazado de rayos en una resolución de 1024 x 1024

3. A pesar de los esfuerzos de Nvidia, colocar todo el código en un solo kernel sigue siendo ineficiente para las GPU. Se llegó a esta conclusión por dos razones principales: primero, la implementación de código abierto con kernel separado en Hydra Renderer ofrece casi el doble de rendimiento que Nvidia RTX para el caso de software puro. En segundo lugar, al comparar dos implementaciones de RTX ligeramente diferentes, se encontraron cambios drásticos en el rendimiento, dependiendo de un ligero cambio en la complejidad de los sombreadores.
4. Se utiliza la creación de trabajo por GPU para rayos. Esta conclusión se confirma por simple observación: al generar una cantidad aleatoria de rayos (de 10 a 40), el rendimiento en [4] se duplicó en comparación con 10 rayos.

Con esto en mente, se deduce que las tecnologías de trazado de rayos han evolucionado significativamente desde sus inicios, con la aparición de soluciones como Nvidia RTX, que ofrecen una aceleración por hardware de manera notable más eficiente. Sin embargo, aún existen áreas de mejora, como la optimización del código en un solo kernel y la gestión de la creación de trabajo por GPU para rayos. Como bien se concluye en [4], la Nvidia RTX es una especie de tecnología "general", orientada a acelerar el acceso aleatorio a la memoria y la distribución irregular del trabajo en las GPU.

III-C. DLSS: Revolucionando el Rendimiento y la Calidad Visual en los Videojuegos

DLSS (Deep Learning Super Sampling) es una técnica de superresolución acelerada por hardware de código cerrado desarrollada por Nvidia [6]. Su propósito general es aumentar la resolución de imágenes renderizadas en baja resolución a una resolución más alta, como 4K u 8K, sin una perdida significativa de calidad visual y mejorando el rendimiento del juego al reducir la carga de trabajo en la GPU. DLSS utiliza redes neuronales entrenadas con imágenes de ultra alta resolución para aprender a generar imágenes de alta calidad a partir de entradas de baja resolución. Durante el entrenamiento, la red recibe imágenes de baja resolución junto con vectores de movimiento y las compara con imágenes de referencia de 16K, ajustando sus pesos para mejorar la precisión. En tiempo real, la red aplica este aprendizaje para reconstruir imágenes de alta resolución desde una entrada de baja resolución [6].

Son muchos los beneficios que trae esta técnica, entre los más destacados podemos mencionar que, aumenta la resolución de imágenes renderizadas internamente en 1080p o 1440p a 4K u 8K sin perdida significativa de calidad visual, disminuye significativamente el aliasing gracias a las muestras de MSAA de 64x por píxel utilizadas durante el entrenamiento, mejora los detalles de textura en resoluciones más bajas, mostrando detalles que solo eran visibles en resoluciones más altas, mantiene la coherencia temporal entre fotogramas, mejora los tiempos de salida de cuadros promedio al reducir la carga en la vram y en la GPU [6].

Figura 4. Captura del videojuego Fortnite, con y sin el uso del DLSS, donde se manifiesta una mejora en sombras, reflejos y texturas. [7]

Cuando DLSS está desactivado, el juego de video o la aplicación se renderiza a la resolución nativa. Esto significa que si configura el juego para ejecutarse en 4K, realmente se renderiza cada cuadro a esa resolución. Con DLSS activado,

el juego de video o la aplicación se renderiza inicialmente a una resolución más baja, como 1080p o 1440p. Esta imagen de menor resolución luego se escala usando algoritmos de aprendizaje profundo a una resolución más alta como 4K u 8K [6].

Al usar DLSS, los usuarios pueden disfrutar de visuales de alta resolución con un mejor rendimiento, lo que lo convierte en una herramienta valiosa para aquellos que buscan aprovechar al máximo su experiencia de juego sin necesitar el hardware más potente disponible.

IV. Conclusiones

Los estudios realizados en [4] demuestran el avance significativo que ha experimentado el trazado de rayos en los últimos años. Sin embargo, aún existen áreas de mejora, como la optimización del código en un solo kernel, la gestión de la creación de trabajo por GPU para rayos y el desarrollo de nuevos algoritmos de búsqueda espacial más eficientes. En este sentido, se espera que Nvidia, en un futuro próximo, continúe innovando en este campo, creando conjuntos de algoritmos y técnicas de aceleración por hardware aún más eficientes para el trazado de rayos.

A pesar de los avances significativos en la optimización del rendimiento de los videojuegos mediante tecnologías como Nvidia RTX, TensorRT y DLSS, aún persisten áreas de mejora cruciales. Es fundamental continuar desarrollando algoritmos más eficientes y técnicas de aceleración por hardware para el trazado de rayos, especialmente en la gestión de la creación de trabajo por GPU y la optimización del código en un solo kernel. Se anticipa que Nvidia seguirá liderando la innovación en este campo, mejorando la calidad visual y el rendimiento de los videojuegos mediante la integración de inteligencia artificial en arquitecturas de computadoras.

Referencias

[1] A. F. Erazo-Luzuriaga, F. M. Ramos-Secaira, P. C. Galarza-Sánchez, and M. F. Boné-Andrade, “La inteligencia artificial aplicada a la optimización de programas informáticos,” *Journal of Economic and Social Science Research*, vol. 3, no. 1, p. 48–63, ene. 2023. [Online]. Available: https://economicsocialresearch.com/index.php/home/article/view/61

[2] Nvidia. (2024) Nvidia tensorrt. [Online]. Available: https://developer.nvidia.com/tensorrt

[3] J. Clayton, “Unlocking peak generations: Tensorrt accelerates ai on rtx pcs and workstations,” *Nvidia*, 2024. [Online]. Available: https://blogs.nvidia.com/blog/ai-decoded-tensorrt-stable-diffusion-automatic1111/

[4] Санжаров, Вадим, V. Sanzharov, Горбоносов, Алексей, A. Gorbonosov, Фролов, Владимир, V. Frolov, Волобой, Алексей, and A. Voloboy, “Examination of the nvidia rtx,” in *ResearchGate*, 11 2019, pp. 7–12.

[5] E. Records. (2022) ¿qué es el trazado de rayos en tiempo real y por qué es importante? [Online]. Available: https://www.unrealengine.com/es-ES/explainers/ray-tracing/what-is-real-time-ray-tracing

[6] B. Sheng, X. Chen, T. Li, T. Ma, Y. Yang, L. Bi, and X. Zhang, “An overview of artificial intelligence in diabetic retinopathy and other ocular diseases,” *Frontiers in Public Health*, vol. 10, 2022. [Online]. Available: https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.971943

[7] Nvidia. (2024) El multiplicador de rendimiento, impulsado por ia. [Online]. Available: https://www.nvidia.com/es-la/geforce/technologies/dlss/