



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Máster Universitario en Software y Sistemas

Trabajo Fin de Máster

**Real-time Anti-Aliasing Techniques in
Game Graphics Processing**

Autor: Zihao Feng

Tutor: Raúl Alonso Raúl

Co-Tutor: José Crespo del Arco

Madrid, Julio 2025

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster

Máster Universitario en Software y Sistemas

Título: Real-time Anti-Aliasing Techniques in Game Graphics Processing

Julio 2025

Autor: Zihao Feng

Tutores:

Raúl Alonso Calvo

Departamento de Lenguajes, Sistemas Informáticos e Ingeniería de Software

José Crespo del Arco

Departamento de Lenguajes, Sistemas Informáticos e Ingeniería de Software

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

El aliasing en el renderizado en tiempo real de los gráficos de los videojuegos siempre ha sido un problema que afecta a la calidad del juego. Los bordes dentados de la imagen no solo hacen que la pantalla parezca menos realista, sino que también pueden provocar molestias visuales al jugador, lo que afecta a la experiencia de juego. Para resolver estos problemas de dentado, se han probado diversas técnicas de antialiasing en el pasado, con el fin de encontrar el mejor equilibrio entre rendimiento y calidad de imagen. En este artículo, exploraremos los principios, los escenarios de aplicación, las ventajas y las desventajas de estas técnicas para proporcionar una solución más eficiente para el renderizado de juegos.

Este estudio utiliza el motor Unreal Engine como plataforma de prueba y emplea las técnicas de antialiasing integradas en el motor, entre las que se incluyen No Antialiasing (NOAA), Fast Approximate Antialiasing (FXAA), Temporal Antialiasing (TAA) y Temporal Super-Resolution (TSR), y realiza pruebas de rendimiento y evaluaciones de la pantalla del juego basadas en el proceso de renderización nativo del motor. Las pruebas incorporan diferentes resoluciones y complejidades de escena para examinar el equilibrio entre la fidelidad de la imagen y la eficiencia computacional.

La metodología experimental utiliza una combinación de análisis de imágenes y comparaciones visuales subjetivas para evaluar el efecto de diversas técnicas de antialiasing del motor Unreal Engine sobre los bordes dentados. El análisis estadístico revela además las compensaciones entre diversos indicadores y las diferencias significativas entre los distintos algoritmos, lo que proporciona un apoyo más convincente a la selección de la tecnología.

Los resultados muestran que FXAA casi no tiene impacto en el rendimiento, pero también es relativamente limitado en la mejora de la calidad de la imagen. Por el contrario, TAA proporciona efectos de antialiasing más completos con un menor coste computacional y puede ser la primera opción para la mayoría de los escenarios de desarrollo. TSR, como tecnología de vanguardia actual, mejora significativamente la claridad de la imagen gracias a su esquema de superresolución, proporcionando el mejor detalle y calidad de bordes a altas resoluciones. Sin embargo, en comparación con otras tecnologías de antialiasing, TSR tiene un coste de rendimiento significativamente mayor a altas resoluciones y la eficiencia de renderizado más baja. Es adecuado para escenarios de aplicación con requisitos de calidad de imagen extremadamente altos y recursos de hardware suficientes.

En resumen, las tecnologías de antialiasing evaluadas en este artículo tienen sus propias ventajas. En el desarrollo real, FXAA es adecuado para dispositivos de gama baja con requisitos de alto rendimiento, mientras que TAA logra un buen equilibrio entre calidad de imagen y rendimiento y es adecuado para la mayoría de los juegos convencionales. TSR, como solución de antialiasing de nueva generación, tiene un efecto significativo en la mejora de la calidad de imagen con el apoyo de dispositivos de hardware. Es especialmente adecuado para escenas de alta resolución con requisitos de alta calidad de imagen. A través del análisis comparativo de este estudio, los desarrolladores pueden elegir la tecnología de antialiasing de forma más razonable según los requisitos del proyecto para mejorar la experiencia general del juego, teniendo en cuenta el rendimiento y la calidad de imagen.

Abstract

Aliasing in real-time rendering of game graphics has always been an issue that affects the quality of the game. Jagged edges of the image not only make the screen look less realistic, but may also make the player feel visually uncomfortable, affecting the gaming experience. In order to solve these jaggedness problems, people have tried various anti-aliasing techniques in the past, trying to find the best balance between performance and image quality. In this article, we will explore the principles, application scenarios, advantages and disadvantages of these techniques to provide a more efficient solution for game rendering.

This study uses the Unreal Engine as the test platform, and uses the built-in antialiasing techniques of the Unreal Engine, including No Antialiasing (NOAA), Fast Approximate Antialiasing (FXAA), Temporal Antialiasing (TAA), and Temporal Super-Resolution (TSR), and conducts performance benchmarking and screen evaluation of the game screen based on the engine's native rendering pipeline. The tests incorporate different resolutions and scene complexities to examine the balance between image fidelity and computational efficiency.

The experimental methodology uses a combination of image analysis and subjective visual comparisons to evaluate the effect of various anti-aliasing techniques of the Unreal Engine on jaggies. Statistical analysis further reveals the trade-offs between various indicators and the significant differences between different algorithms, providing more convincing data support for technology selection.

The results show that FXAA has almost no impact on performance, but is also relatively limited in improving image quality. In contrast, TAA provides more comprehensive anti-aliasing effects at a lower computational cost and can be the first choice for most development scenarios. TSR, as the current state-of-the-art technology, significantly improves the image clarity by virtue of its super-resolution scheme, providing the best detail and edge quality at high resolutions. However, compared with other anti-aliasing technologies, TSR has a significantly higher performance cost at high resolutions and the lowest rendering efficiency. It is suitable for application scenarios with extremely high image quality requirements and sufficient hardware resources.

In summary, the antialiasing technologies evaluated in this paper each have their own advantages. In actual development, FXAA is suitable for low-end devices with high performance requirements, while TAA strikes a good balance between image quality and performance and is suitable for most mainstream games. TSR, as a new generation of anti-aliasing solution, has a significant effect on improving image quality with the support of hardware devices. It is especially suitable for high-resolution scenes with high image quality requirements. Through the comparative analysis in this study, developers can choose antialiasing technology more reasonably according to the project requirements to improve the overall game experience, while taking into account the performance and image quality.

Contents

List of Abbreviations	v
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Objectives.....	2
2 Background	4
2.1 Basic Principles of Anti-Aliasing Techniques	4
2.2 History and Development of Anti-Aliasing Technology	6
2.3 Overview of Anti-Aliasing Techniques in Games	6
2.3.1 Spatial Domain Anti-Aliasing Methods	7
2.3.2 Temporal Domain Anti-Aliasing Methods	8
2.3.3 Post-processing Anti-Aliasing Methods.....	10
2.3.4 Super-Resolution Methods	12
2.4 Challenges of Real-Time Anti-Aliasing Techniques.....	13
3 Evaluation Methods and Image Quality Metrics	15
3.1 Evaluation Objectives and Motivation for Metric Selection	15
3.2 Peak Signal-to-Noise Ratio (PSNR)	16
3.3 Structural Similarity (SSIM)	16
3.4 Edge Preservation.....	17
3.5 FPS	18
3.6 GPU Time	18
3.7 Evaluation Workflow and Implementation Tools	18
4 Experimental Design	20
4.1 Experiment Objectives and Approach	20
4.2 Experimental Platform and Environment.....	20
4.3 Experimental scene design	21
4.4 Anti-Aliasing Techniques Evaluation Method.....	23
4.4.1 Image Quality Assessment Metrics	23
4.4.2 Image Performance Evaluation Metrics	23
4.4.3 Reference Image Selection Strategy	23
4.4.4 Image Resolution and Multi-Group Data Evaluation	24
4.5 Image Quality and Performance Evaluation Workflow.....	24
4.6 Experimental data visualization	27
4.7 Statistical analysis methods	27
5 Results and Analysis	29
5.1 Image Quality Metrics Analysis	29
5.2 Performance Analysis	37
5.3 Statistical Analysis	42

5.3.1	Descriptive Statistics.....	42
5.3.2	Normality Testing.....	44
5.3.3	One-way ANOVA and T-Test.....	44
5.3.4	Non-Parametric Testing (Kruskal-Wallis).....	46
5.3.5	Correlation.....	47
5.4	Subjective Visual Evaluation	49
5.4.1	Visual comparison of different anti-aliasing techniques- Part 1 ..	49
5.4.2	Visual comparison of different anti-aliasing techniques- Part 2..	53
5.5	Discussion	55
6	Conclusion.....	57
6.1	Research Summary	57
6.2	Limitations and future research directions	58
6.3	Future Work Prospects.....	59
7	Personal Reflections	60
	Bibliography	61
	Anexos.....	64

List of Abbreviations

SSAA Super-Sampling Anti-Aliasing

MSAA Multisampling Anti-Aliasing

MLAA Morphological Anti-Aliasing

NOAA No Anti-Aliasing

FXAA Fast Approximate Anti-Aliasing

TAA Temporal Anti-Aliasing

TSR Time Super Resolution

DLSS Deep Learning Super Sampling

PSNR Peak Signal-to-Noise Ratio

SSIM Structural Similarity Index

1 Introduction

1.1 Context

With the rapid advancement of hardware technology in recent years, the increasing complexity of game graphics, and the growing demands of players for visual experiences, aliasing has remained a long-standing issue for both developers and players in real-time game graphics rendering. It primarily manifests in three forms: staircase artifacts on geometric edges, flickering of high-frequency textures, and temporal instability in dynamic scenes. While traditional super-sampling anti-aliasing (SSAA) offers excellent image quality, it is so computationally costly that it is not acceptable in modern real-time rendering. Real-time anti-aliasing techniques that use post-processing and temporal accumulation have therefore become widespread in gaming.

This study focuses on three mainstream real-time anti-aliasing techniques in the Unreal Engine: FXAA, TAA, and TSR. Although deep learning super sampling techniques (such as DLSS) have emerged in the industry, their reliance on high-end hardware (such as NVIDIA RTX high-end graphics cards) limits their universality. In contrast, FXAA, TAA, and TSR, as engine-native anti-aliasing techniques, have broader applicability and are more suitable for deployment in cross-platform projects. However, these technologies also face many challenges in practical applications:

1. As a representative of post-processing anti-aliasing, FXAA achieves rapid aliasing suppression through full-screen edge detection and blurring operations. However, its inherent over-blurring issue often leads to the loss of texture details in images, particularly in high-frequency detail scenes such as vegetation and hair.
2. TAA can clearly increase temporal steadiness and reduce flicker in dynamic scenes via utilization of multi-frame history information. It can, however, cause ghosting if motion vectors it relies on are not accurate enough, something that happens mostly in rapid-motion scenes and dynamic lighting situations.
3. TSR, an Unreal Engine 5 temporal super-resolution technology, is an extension of resolution scaling functionality on top of TAA, theoretically enabling reconstruction of high-quality images at sub-TAA rendering resolutions. Its algorithm complexity is, however, towards the higher side, requiring additional memory bandwidth and computational power, thus having the potential to cause stuttering on less-expensive hardware platforms.

In recent years, with the development of AI, academic research on AI-driven anti-aliasing solutions such as DLSS and FSR has increased. However, mainstream games and game development still heavily rely on native AA technologies within game engines. Systematic comparative analyses of native AA technologies within game engines remain insufficient, and developers often rely on empirical choices without quantitative data support. This study designs controlled experimental scenarios, combining objective image quality metrics (PSNR, SSIM, edge retention) with subjective user evaluations, to provide game developers with empirical evidence for anti-aliasing technologies tailored to different project needs, revealing the advantages and limitations of various anti-aliasing techniques in practical applications.

1.2 Motivation

Due to the extremely rapid growth in the gaming industry in recent years, technological advances in graphics output and processor technology have prompted ever-more demanding requirements for immersive gaming from gamers. In turn, anti-aliasing technology, which can take games to an entirely new level of realism, is becoming an integral part of real-time rendering systems. Nonetheless, in current mainstream anti-aliasing systems, several challenges still exist in practical usage: FXAA's over-blurring causes detail in textures to be lost, TAA tends to cause ghosting during rapid-screen movements, and TSR, where anti-aliasing performance is at its strongest, needs hardware performance to be at least in certain specifications. While AI-powered anti-aliasing systems (such as NVIDIA's DLSS) do exist, due to their extremely high cost and dependence on GPU hardware, even widespread usage in such cutting-edge anti-aliasing systems in markets still isn't possible, so not everyone can ever use current systems at launch.

At present, it is common for players and developers to mostly rely on experience or word-of-mouth from comparison videos in selecting anti-aliasing technologies, whereas systematic experimental performances have not been routinely made and quantitatively analyzed. This situation may make it difficult for developers to choose the optimal solution tailored to specific project requirements (such as cross-platform compatibility, high-fidelity visual effects, or performance optimization), while players may also be unclear about which anti-aliasing technology is best suited for their computers. Therefore, this study aims to fill the existing research gap through scientific experimental design and multi-angle evaluation, providing developers with data-driven insights for selecting anti-aliasing methods and offering gamers a basic reference for choosing anti-aliasing technologies.

1.3 Objectives

The overall goal of this study is to build and verify a feasible real-time anti-aliasing technology selection and evaluation framework, and provide data-driven decision support for game graphics developers by comparing the image quality and performance overhead of multiple mainstream algorithms at different resolutions and scenes. The specific research objectives are as follows:

1. Construction of image quality evaluation system

Select three objective indicators, PSNR, SSIM and edge preservation, design an automated comparison script, and systematically quantify the static image quality performance of the three anti-aliasing algorithms FXAA, TAA, and TSR.

2. Performance test process design

Based on the Unreal Engine platform, use the stat fps and stat unit tools to collect and count the average FPS and GPU rendering time of each algorithm at 4K resolution, and build a performance test template.

3. Resolution and scene applicability analysis

In at least three representative scenes (low, medium, and high complexity) and 1K/4K resolutions, cross-compare the quality-performance performance of the algorithm to reveal its resolution dependence and scene adaptability.

4. Comprehensive trade-offs and visualization support

In order to more comprehensively compare the advantages and disadvantages of different anti-aliasing technologies, this article not only analyzes various image quality and performance indicators from a single dimension, but also combines a variety of data visualization methods such as bar charts, radar charts, and heat maps to present the performance of different algorithms on various indicators in an intuitive graphical manner. Through visual analysis, not only can the advantages and disadvantages of various anti-aliasing technologies in image quality (such as PSNR, SSIM, edge retention) and performance (such as FPS, GPU rendering time) be clearly compared, but also the performance trends of different algorithms under different scene complexities can be revealed at a glance, providing a more scientific and transparent decision-making basis for the subsequent technology selection chapters.

5. Practical selection guide output

Based on the rigorous experimental process and statistical analysis results of this study, we finally summarized the anti-aliasing technology adaptation recommendations for different application scenarios: In high-end scenes with high resolution or extremely high image quality requirements, which anti-aliasing technology is recommended to obtain the best detail performance and edge quality; for mainstream game development and daily interactive applications, which anti-aliasing technology can achieve a better balance between image quality and performance; and on platforms with extreme rendering performance requirements and limited hardware resources, which anti-aliasing technology is a more practical solution.

2 Background

This chapter provides an introduction to the basic principles of anti-aliasing, reviews the development history of anti-aliasing technology, and categorizes it into spatial anti-aliasing, temporal anti-aliasing, and super-resolution anti-aliasing. It briefly introduces several anti-aliasing techniques used in real-time rendering in games, and finally discusses the challenges currently faced in the development of anti-aliasing technology.

2.1 Basic Principles of Anti-Aliasing Techniques

Aliasing[1] is a very common visual problem in digital images and real-time graphics processing. When rendering high-frequency details (such as sharp edges, textures, or distant objects), sampling at a resolution lower than the Nyquist frequency can lead to visible jagged or "staircase" patterns, such as jagged edges, or temporal flicker, collectively known as aliasing.

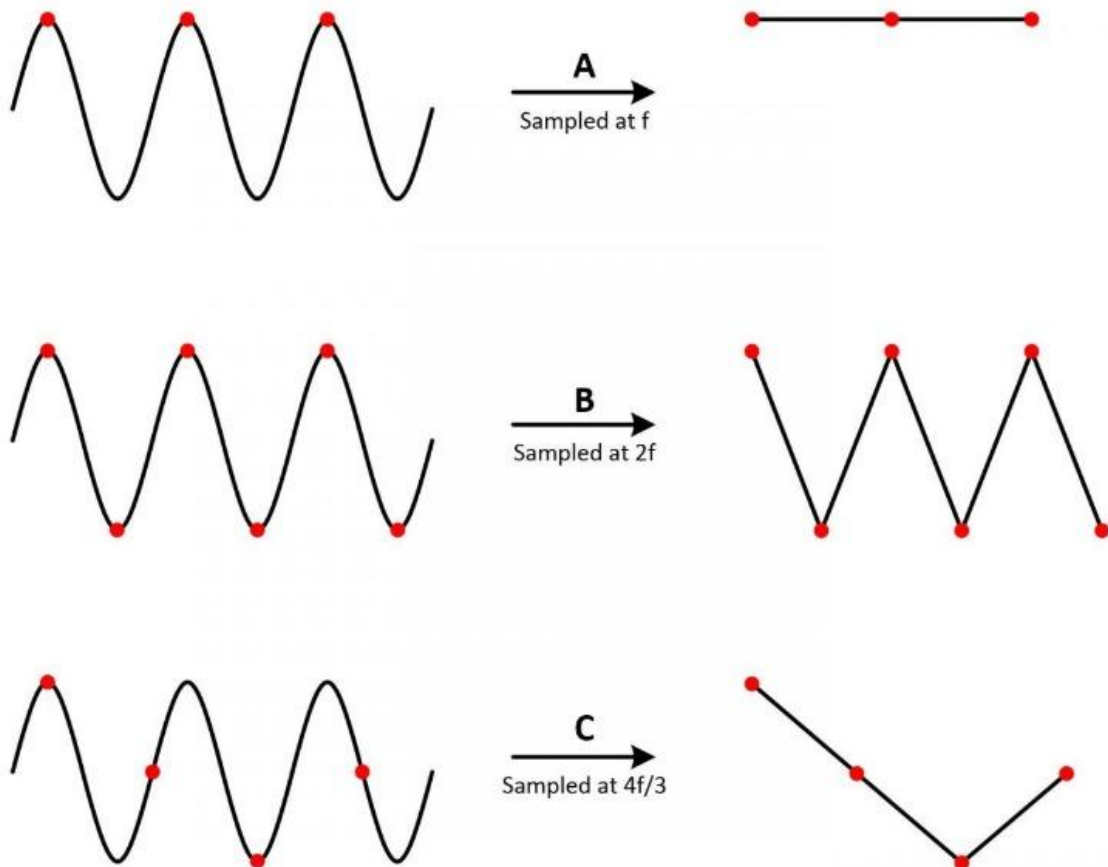


Figure 2.1 Effect of Sampling Rate on Signal Reconstruction

According to the Nyquist-Shannon sampling theorem, in order to accurately reconstruct a continuous signal from samples without aliasing, the sampling frequency must be at least twice the highest frequency component of the signal. This threshold is called the Nyquist frequency.

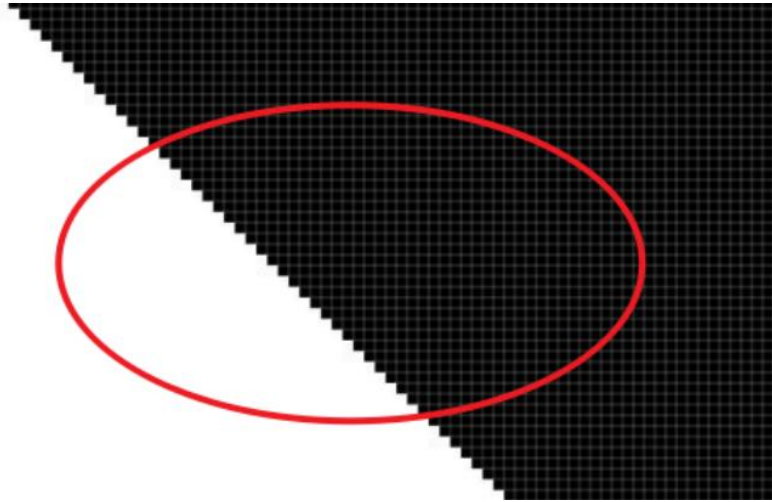


Figure 2.2 Aliasing image

When this phenomenon appears on a computer screen, the image will appear unrealistic and may even make people feel uncomfortable. In order to solve this problem, people invented anti-aliasing technology. From the perspective of the Nyquist frequency, anti-aliasing technology can be understood as a method of increasing the sampling resolution or suppressing the frequency that would otherwise cause aliasing due to insufficient sampling. The core of anti-aliasing is to reduce or eliminate such aliasing phenomena by smoothing object contours and optimizing the visual transition between adjacent pixels, thereby improving the visual effects and realism of computer images. Modern anti-aliasing algorithms are mainly divided into four categories: spatial anti-aliasing, temporal anti-aliasing[2], post-processing anti-aliasing, and AI-based super-resolution anti-aliasing:

Spatial sampling: This type of technology suppresses aliasing at the source by supersampling or multisampling high-frequency geometry or texture information before the rasterization stage, sampling the continuous signal at a higher resolution, and then downsampling it back to the target resolution. For example, SSAA, MSAA.

Temporal sampling anti-aliasing techniques: By combining information from past and current frames, pixels are sampled and blended to smooth edges and reduce jaggedness. For example, TAA.

Post-processing anti-aliasing techniques (image-based filtering): Post-processing anti-aliasing is a secondary filtering performed on the image after rendering is completed. It quickly analyzes the amplitude, gradient or depth information of edge pixels, and then blends the colors of adjacent pixels to smooth out rough jagged edges. For example, FXAA, MLAA.

AI-based super-resolution: Using neural networks to render images at a lower resolution first[3], then upscale them to the target resolution using super-resolution technology, and applying anti-aliasing algorithms during the upscaling process to reduce jaggedness, thereby improving rendering performance while ensuring image quality[4]. For example, DLSS, TSR.

Anti-aliasing technology aims to minimize the jagged edge effects of object boundaries during graphics rendering, and improves the smoothness of the picture through algorithmic optimization, resulting in more realistic visual effects.

2.2 History and Development of Anti-Aliasing Technology

Ever since the display technology emerged, the need for image quality among human beings has constantly advanced step by step, and such high demands for image quality have spawned the continuous iteration of anti-aliasing technology accordingly. The developmental history of anti-aliasing technology is also that of the persistent co-existence between image quality and compromise towards performance in computer graphics. Human beings used super-sampling anti-aliasing technologies[5] in the past in order to obtain nearly ideal anti-aliasing effect. The thinking was simple and direct: sample the image at high resolution and down-size it to the resolution desired. However, such solution bears heavy calculation cost and neither scales nor sustains itself, and that gives birth to MSAA[6].

Developers quickly developed an optimized version of SSAA, which only performs multiple sampling on the edges of polygons. This method significantly reduces the computational load on the graphics card while still delivering excellent visual performance, making it widely popular upon its release. However, MSAA also has its limitations. For semi-transparent objects like blades of grass or wire mesh, whose edges are fuzzy or very complex, the shortcomings of MSAA are apparent. MSAA itself actually represents the compromise regarding quality and performance. It merely optimizes the sampling only at the geometry edges level instead of at the level of the entire image, greatly reducing the calculation amount and ensuring an image quality. In 2010, with the diverse increasing requirement for mobile, TV and PCs' real-time rendering, even with the texture blur's restriction, post-processing technologies such as FXAA and SMAA triumphed due to the support for real-time effect and low overhead. Such developmental trajectory is revealing not only how rendering algorithms are boosted by technological progress at the hardware level, but instead exposing the eternal pursuit for high-quality and efficient visual solution at the level of real-time graphics rendering.

However, these methods either fail to effectively eliminate flickering, leaving temporal distortion defects, or take too long to process, resulting in obvious defects in the final rendering results. In recent years, machine learning-based technologies, such as deep learning sampling (DLSS)[7] and frame synchronization rendering (FSR), have introduced anti-aliasing technology into the field of intelligent image reconstruction, which can effectively improve image quality. These technologies successfully combine inter-frame anti-aliasing with deep learning and achieve excellent anti-aliasing effects. However, due to hardware and architectural limitations, only limited high-end graphics cards currently support this technology. With the continuous development of anti-aliasing technology and people's continuous pursuit of image quality realism, these continuously improving anti-aliasing technologies can not only provide better image quality, but also gradually meet the requirements of low-end systems for real-time processing, significantly improving user experience. From the initial super sampling to the current integration of deep learning and anti-aliasing technology, the continuous development of anti-aliasing technology has brought people an increasingly real and natural visual experience.

2.3 Overview of Anti-Aliasing Techniques in Games

As the quality of real-time graphics rendering improves, anti-aliasing technology[8] is also evolving according to the types of modern games and game engines. According to different processing methods, the current mainstream anti-aliasing methods are roughly divided into three types: spatial anti-aliasing, temporal anti-aliasing, post-processing anti-aliasing and super-resolution anti-aliasing. The following are explained separately:

2.3.1 Spatial Domain Anti-Aliasing Methods

The principle of spatial anti-aliasing[9] is mainly derived from the sampling theorem. Anti-aliasing effects are produced by increasing the image resolution or sampling rate. It is generally implemented by pre-calculating cache textures that are several times the screen resolution. When rendering, the color value of each pixel is calculated using high resolution. When displaying, the high-resolution image is resampled to the target resolution, and the color values of multiple sample points are mixed to produce smooth pixel color values, thereby achieving anti-aliasing effects. Common methods include:

- SSAA[10] (Super Sampling Anti-Aliasing): One of the earliest types of anti-aliasing, it is accomplished by always rasterizing at a resolution that is higher than that desired and then scaling down. It generates the best quality image but is too computationally expensive and appears increasingly rarely during real time rendering.

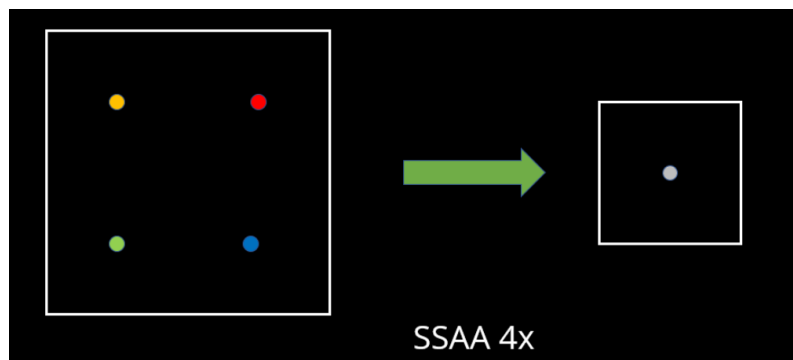


Figure 2.3.1 $4\times$ SSAA sampling

The 4X SSAA algorithm uses a frame buffer and a depth buffer to temporarily cache the rendered image and the depth value of the camera. These buffers are 4 times the size of the screen resolution. When transferring the image from the frame buffer to the screen, the average of the 4 sub-pixel colors in the buffer is the screen pixel color, and the same is true for the depth buffer. Because each frame needs to be calculated and 4 times the pixel and depth information of the screen resolution must be stored, this technology will have a considerable impact on rendering performance.

- MSAA (Multi sample Anti-Aliasing) is a technique that only smooths parts of a frame. MSAA looks primarily at places where aliasing is likely to occur, such as the edges of geometry. Where aliasing issues may occur on an edge, MSAA manipulates its color so that it is between the colors of the two pixels that make up the edge. The dithering effect gives the illusion of smoother edges.

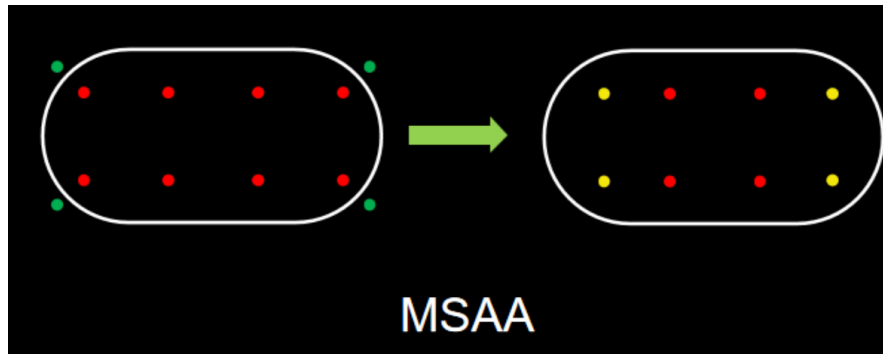


Figure 2.3.2 4×MSAA sampling

In 4×MSAA, the rasterization stage predefines four sub-sampling points inside each pixel, and performs depth and stencil tests on these four points to generate a coverage mask; when at least one sub-sample is covered by geometry, the fragment shader is called only once at the center of the pixel (or a preset position), and the calculated color is then copied to all covered sub-samples; finally, in the Resolve stage, the colors of the four sub-samples are simply averaged or weighted merged to output smooth anti-aliased pixels, thereby greatly reducing the number of shader calls and memory bandwidth overhead while retaining edge geometry details.

The quality of MSAA depends on how many samples it uses to blend the colors along the edges it determines to be aliased. More samples result in better visual quality, but at the expense of more GPU processing.

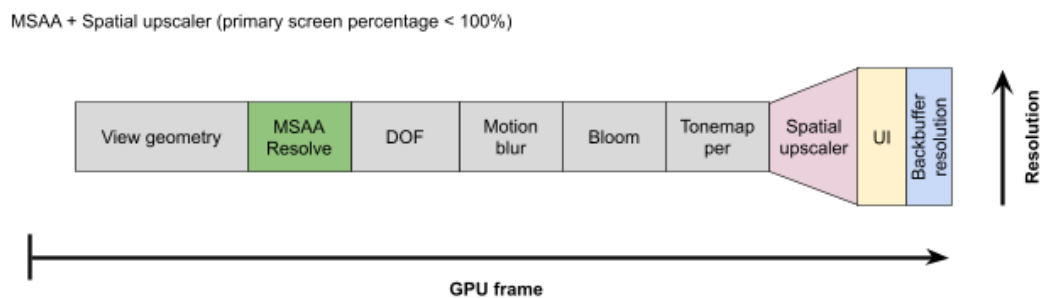


Figure 2.3.3 MSAA rendering chain flow chart

In the rendering chain, MSAA primarily addresses geometry aliasing along edges before the spatial up-sampler. Aliasing in materials, textures, and transparent surfaces is not affected by MSAA.

This type of spatial anti-aliasing technology performs sub-sampling or depth sampling, or stencil multi-sampling on each pixel during the rasterization stage. It can accurately process edges based on real geometric information, eliminate geometric aliasing and retain rich details without the need for additional post-processing and is well optimized at the hardware level. However, this "pre-processing approach" will significantly increase memory bandwidth and shader calculation overhead, especially SSAA will directly increase the rendering resolution by multiples. Although MSAA reduces some costs by merging shading calls, it is still difficult to achieve the best compromise between high quality and balanced real-time performance due to its poor compatibility with the deferred rendering pipeline and limited support for transparent and multiple refraction scenes.

2.3.2 Temporal Domain Anti-Aliasing Methods

Temporal anti-aliasing technology utilizes information from the current frame and the previous frame, combined with motion vectors, to perform pixel interpolation and fusion, thereby eliminating jagged edges and flickering.

- **TAA (Temporal Anti-Aliasing):** is one of the most widely used temporal anti-aliasing techniques today. It fuses pixel data from historical frames with the current frame through reprojection, effectively reducing aliasing and image flickering. TAA performs well in terms of stability and is particularly suitable for scenes with slow or medium-speed camera movements.

The basic principle of TAA (temporal anti-aliasing) is to sample and fuse pixel information between multiple consecutive frames to effectively eliminate aliasing and image jitter. Specifically, TAA first collects pixel data from the current frame and the previous frame or multiple historical frames, calculates the motion vector of the pixel, accurately maps the pixel of the previous frame to the position of the current frame, and uses motion compensation technology to smooth pixel changes. In order to improve image continuity, TAA will introduce a certain amount of motion blur during the synthesis process, and reduce the image quality loss caused by blur through post-processing steps such as sharpening and contrast enhancement. In the end, TAA can output a high-quality picture with smooth edges, significantly reduced aliasing and flickering. As shown in the figure below:

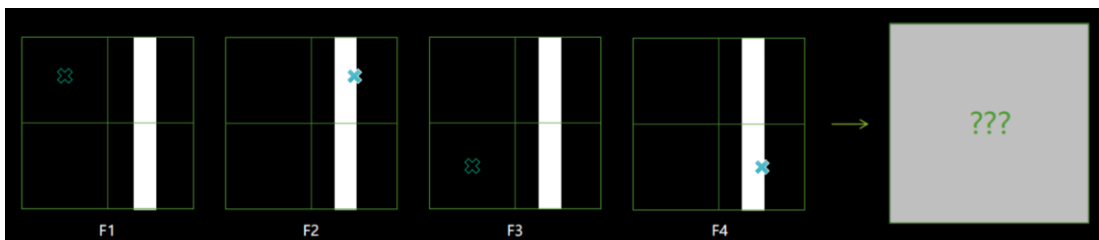


Figure 2.3.4 Example of TAA sampling process

The time-resistant rendering chain will be as follows:

Spatial & temporal upsample (primary screen percentage < 100%)

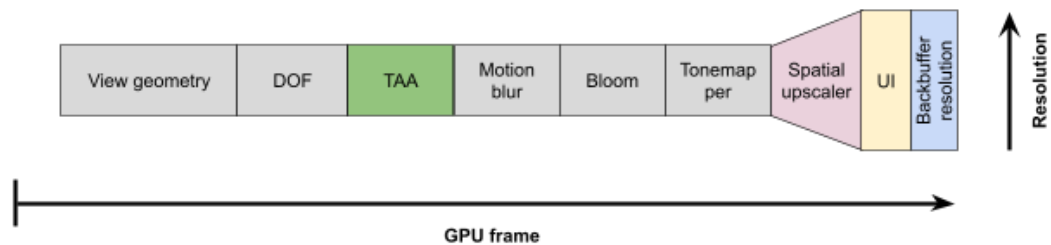


Figure 2.3.5 TAA rendering chain flow chart

- **TXAA:** NVIDIA's TXAA anti-aliasing technology is actually a combination of TAA + MSAA (Multi-Sample Anti-Aliasing). It uses MSAA on deferred rendering by introducing additional depth information. TXAA is specifically designed to be directly integrated into the game engine. TXAA combines the powerful capabilities of MSAA with complex high-quality filters similar to

those used in CG movies. It can also jitter the sampling position between frames to obtain higher image quality.

This type of temporal domain anti-aliasing method effectively suppresses aliasing and flickering problems by fusing the current frame with multiple frames of historical image information and using pixel motion compensation, greatly improving the smoothness of the image and visual stability in dynamic scenes. Its main advantage is that it can greatly improve edge aliasing, flickering and other artifacts at a low performance cost, and has good adaptability to complex dynamic scenes and high-resolution images. In addition, TAA can handle scenes with rich details well and has good compatibility and practicality on most mainstream hardware platforms.

However, TAA also has some shortcomings that cannot be ignored.[11] Due to the introduction of historical frames and imperfect motion vectors, TAA is prone to "ghosting" or smearing, especially in high-speed motion or drastic scene changes. In addition, TAA may cause a certain degree of image blurring in edge areas and high-frequency details, resulting in loss of details. If the motion estimation is inaccurate, problems such as image afterimage or jitter may also occur. To solve these defects, post-processing methods such as sharpening are usually required to repair them.

In general, TAA achieves a good balance between performance and image quality in modern game rendering, but its side effects such as blur and ghosting also require developers to make comprehensive trade-offs based on project needs.

2.3.3 Post-processing Anti-Aliasing Methods

The characteristic of post-processing anti-aliasing algorithm is that it can be processed after rendering is completed, using some filtering techniques in image processing to perform anti-aliasing and edge smoothing operations on the image to improve the image quality. Compared with other types of anti-aliasing algorithms, post-processing anti-aliasing algorithms do not require modifications to various stages in the rendering pipeline, nor do they increase the hardware burden, and have strong compatibility. Mainstream post-processing anti-aliasing algorithms include FXAA, MLAA, SMAA, etc.[12]

- FXAA[13] (Fast Approximate Anti-Aliasing): It is a quick post-processing algorithm that doesn't need multiple samples. Instead, it identifies edges by tracking the brightness gradients of image pixels and blurs. FXAA is highly efficient with low overhead for rendering, which is why it is commonly seen on devices that are mid-to-low-end. Its downside, however, its disadvantage is that compared with other anti-aliasing techniques, it is easy to cause overall blurring of the image and loss of details, and the final image may lose a certain degree of fidelity.

Spatial & temporal upsample (primary screen percentage < 100%)

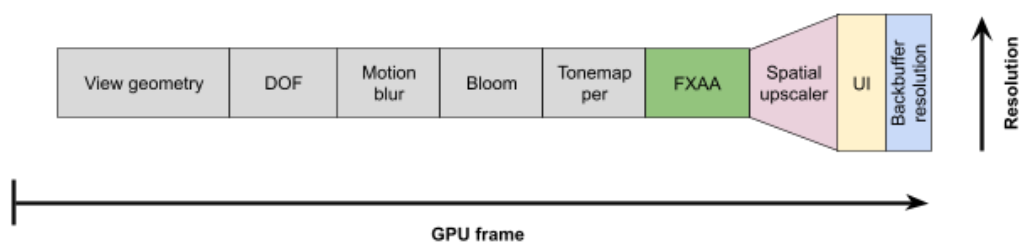


Figure 2.3.6 FXAA rendering chain flow chart

In the spatial and temporal up-sampling chain, the spatial up-sampler occurs at the end of the post-processing chain at the primary screen percentage.

- MLAA[14] (Morphological Anti-Aliasing) is a post-processing anti-aliasing algorithm based on screen space. Its core idea is to first detect high-contrast jagged edges in the final rendered color buffer (usually using Sobel, Luma or depth gradient), then match the predefined blending template according to the morphological features of the pixel neighborhood (horizontal, vertical or diagonal edges), perform pixel-level color blending and interpolation on these edge pixels, and finally output a smooth result. Unlike MSAA, MLAA does not rely on geometric information or multi-sampling hardware support, and can be seamlessly applied to deferred rendering pipelines. The main overhead comes from edge detection and template matching; but it is also prone to blurring details, over-smoothing on complex textures, and is not as good as Temporal AA in handling "flickering" in dynamic scenes.

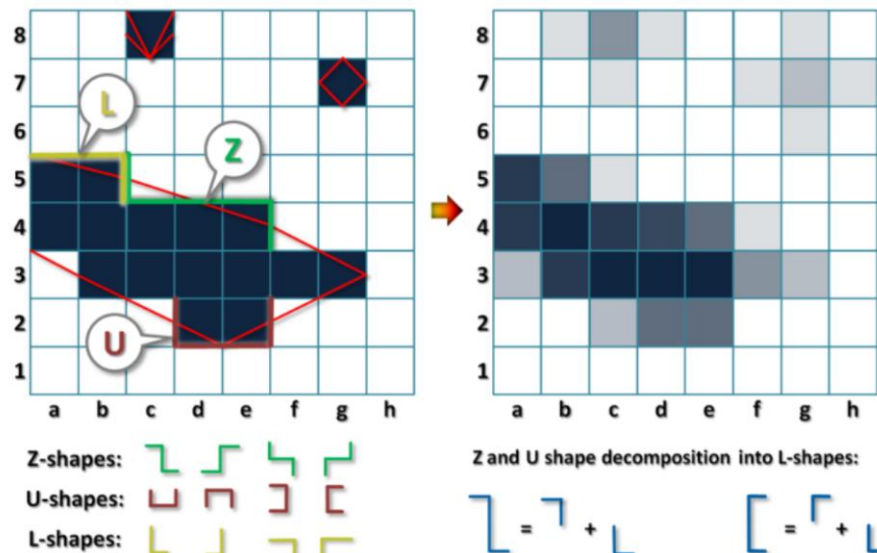


Figure 2.3.7 FXAA rendering chain flow chart

As shown in Figure 2.3.7, the idea of the MLAA algorithm[15] is mainly based on the principles of morphology. The algorithm first detects information such as the brightness, color, depth or normal of the object edge, and then divides the detected edges into three shapes of Z, U, and L for pattern recognition and saves them. Next, it re-vectorizes the edge shape and calculates the coverage area to determine the blending weight. Finally, the edge is blended with the background color to achieve an anti-aliasing effect.

This type of post-processing anti-aliasing[16] method only performs screen-space filtering on the final frame buffer once, and thus has the advantages of low performance overhead, easy integration, compatibility with deferred rendering, and the ability to smooth transparency and particle effects. However, since edge detection and template matching are only based on pixel brightness or color gradients, it is very easy to produce over-smoothing and blurring, misjudgment of edges on small details or high-frequency textures, and cannot reconstruct true sub-pixel geometric boundaries, so that jagged edges or false edges will still remain on extremely fine diagonal lines or complex patterns.

2.3.4 Super-Resolution Methods

Super-resolution anti-aliasing is a rapidly developing field in image reconstruction. It combines image scaling with anti-aliasing processes, rendering at low resolution and then upscaling the image to high-quality output through complex algorithms while simultaneously performing anti-aliasing. Representative methods include:

- DLSS (Deep Learning Super Sampling): NVIDIA's technology that applies image inference and reconstruction with the use of neural networks, achieving superb sharpness and temporal stability. Its anti-aliasing quality is equal to or superior to that of common TAA, yet with notably low GPU loading. Nevertheless, its platform compatibility is restricted due to the need for specialized hardware (like Tensor Cores) and learning data.
- AMD FSR (FidelityFX Super Resolution): Open-source spatial super-resolution algorithm with wider compatibility, compatible with mainstream GPUs, but usually slightly worse image quality compared with DLSS.
- XeSS (Xe Super Sampling): Intel's deep learning-based super-sampling[17] tech that was brought out with the aim to improve visual quality and gameplay performance. It first lowers the rendering resolution of the game and then applies AI-based algorithms to scale the image at a greater resolution, enhancing frame rates with minimal image quality compromise.
- TSR[18] (Temporal Super Resolution): Epic Games' anti-aliasing + super-resolution solution for Unreal Engine 5 combines TAA's temporal reconstruction with high-quality sharpening to generate ultra-high-definition visuals at resolutions higher than the original. TSR does not rely on AI accelerators and offers excellent versatility and image quality. With TSR, you can achieve image quality close to 4K resolution while also halving GPU frame times.

Temporal Super Resolution has the following properties:

- Rendered frames are close to native 4K quality with input resolutions as low as 1080p.
- Compared to Unreal Engine 4's default anti-aliasing method, i.e. temporal anti-aliasing, there are fewer visible "ghosting" artifacts on high-frequency backgrounds.
- Reduced flickering issues on highly complex geometry
- Support for dynamic resolution scaling on console platforms.

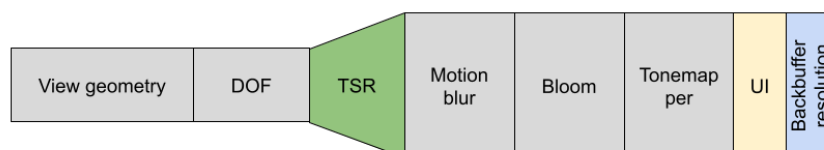


Figure 2.3.8 TSR rendering chain flow chart

In the rendering pipeline, temporal super-resolution occurs after depth of field, and all subsequent content undergoes resolution modification, such as motion blur, floodlighting, etc.

Super-resolution anti-aliasing is increasingly important in modern 3A game development, especially in high-resolution, high-refresh-rate rendering scenarios, where it has become a key technology for balancing image quality and performance[19]. This study uses TSR technology as a representative of super-resolution anti-aliasing techniques to investigate the image quality and performance data after processing with such anti-aliasing technologies.

2.4 Challenges of Real-Time Anti-Aliasing Techniques

Real-time anti-aliasing technology, as a vital branch of computer graphics, has undergone decades-long steady innovation and development. After the emergence of the early 1990s' Multi-Sample Anti-Aliasing (MSAA), the technology significantly improved the edge smoothness of the game images and provided players with a more natural visual effect. Subsequent innovations such as Fast Approximate Anti-Aliasing (FXAA) and Temporal Anti-Aliasing (TAA) balanced the performance and picture quality further and particularly demonstrated their vigor on resource-constrained hardware platforms. In the recent years, high-order methods based on deep learning and ray tracing (such as Temporal Super Resolution, TSR) and NVIDIA's DLSS (Deep Learning Super Sampling) have propelled the development of anti-aliasing technology into higher image quality and efficiency. However, even amidst the fantastic development achieved, the current real-time anti-aliasing methods still have many long-term and short-term challenges within the fields of contemporary game production[20].

Firstly, the performance and picture quality sacrifice remains the core issue: high sampling rates (such as MSAA 8x or higher) or sophisticated post-processing processes (such as TAA) can effectively inhibit the aliasing effect, but increase the computational cost on the GPU heavily, particularly under high-resolution rendering (such as 4K or 8K resolution), where the loss of the frame rate directly affects the playability.

Secondly, the non-adaptability on the dynamics of a scene is another major issue. The traditional MSAA shows optimality for static geometric edges but performs poorly on motion blur, alpha textures (such as leaves or hairs or hair), or moving targets. TAA increases the picture quality based on the integration with time, but often pays the visual continuity for ghosting effect. In addition, the algorithmical complexity and compatibility on hardware further limit the scope of the technology. Advanced anti-aliasing technology generally relies on special hardware (such as NVIDIA RTX series Tensor Cores), which leads to compatibility problems for legacy GPUs or cross-platform developing (e.g., Nintendo Switch and PlayStation 5).

Besides, it's also difficult to achieve an optimal balance between edge and texture detail. FXAA blurs edges with a low cost but generally introduces too much blur, and MSAA can't effectively deal with the issue on the problem of

alias on the transparent surfaces, which leads to non-constant image quality. Besides, the compromise between real-time ability and latency is particularly critical on highly interactive games (such as VR or competition games that require the player's high-speed input and response), where the dynamical integration on TAA and TSR can lead to input latency and affect the players' response time.

And lastly, the expensive cost on developing and optimizing leads to difficulties for medium and small-scale developing teams on utilizing the advanced technology widely, which makes the application limited among indie games or projects on limited budgets. In a word, although there have been extraordinary accomplishments on theory and practice, real-time anti-aliasing technology still undergoes numerous bottlenecks on the issue on optimizing the performance, dynamical adaptability, compatibility, consistency on the image quality, latency control, and developing accessibility. These issues not only pose the academy with vast research opportunities but also set the future developing directions on the technology for the graphic system on the game.

3 Evaluation Methods and Image Quality Metrics

This chapter introduces image quality and performance metrics used to evaluate the effectiveness of different anti-aliasing techniques. To objectively and systematically compare the actual performance of the three anti-aliasing algorithms—FXAA, TAA, and TSR. This study establishes an evaluation framework based on image analysis and performance monitoring, covering both subjective and objective dimensions, as well as quality and efficiency aspects.

First, this chapter will provide a detailed introduction to the three classic objective metrics used in image quality evaluation: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Edge Preservation. These metrics assess the effectiveness of anti-aliasing techniques from three perspectives: pixel differences, structural consistency, and edge sharpness.

Second, to measure the impact of different anti-aliasing techniques on rendering performance, this paper also introduces two commonly used performance metrics: frames per second (FPS) and GPU rendering time (GPU Time), which are used to evaluate the resource consumption and efficiency of different algorithms in actual operation.

Finally, this chapter will also describe the technical tools (such as Python + OpenCV + skimage), data processing methods, and visualization techniques used in the entire evaluation process to ensure the reproducibility, quantifiability, and scientific rigor of the evaluation process.

3.1 Evaluation Objectives and Motivation for Metric Selection

The goal of antialiasing technology is to reduce and eliminate jagged edges and artifacts in images to enhance visual experience and image quality, especially in dynamic rendering and real-time game scenes. The evaluation of game image quality is a complex process and is generally performed in both subjective and objective ways. The subjective approach is based on the information collected from the human eye senses, i.e., the visual effects, the effects of jaggedness in the image are present at the edges of the image objects, which are part of the image details, and a single subjective evaluation is difficult to reflect the merits of their performances and implement comparisons. Therefore, it is important to quantitatively evaluate the ability of these techniques in terms of image fidelity, structure restoration and edge preservation using multiple objective image quality metrics.

In this study, three evaluation metrics are used to assess the performance of each antialiasing technique at 1K and 4K: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Edge Preservation[21]. PSNR measures the overall error at the pixel level, SSIM focuses on the retention of structural information in the image, and Edge Preservation focuses more on the sensitivity of the human eye to clarity and edge sharpness. Edge Preservation focuses on the sensitivity of the human eye to sharpness and clarity[22]. These three metrics can quantify the details of anti-aliased images from different angles and

complement each other. By combining these metrics, the actual effect of each anti-aliasing technology can be compared more objectively and comprehensively.

3.2 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is one of the most commonly used metrics for evaluating image quality. It quantifies the degree of image distortion by calculating the mean squared error (MSE) between the original image and the processed image. The calculation formula is shown in the figure below:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

where MAX represents the maximum possible value of an image pixel (if each sample point is represented by 8 bits, then it is 255), and MSE is the mean squared error.

The PSNR values for the original image and the processed anti-aliasing image typically range between 30 dB and 50 dB. A higher PSNR value indicates a smaller difference between the two images and better image quality.

When PSNR is between 20 dB and 30 dB, the human eye can perceive differences in the image.

When PSNR exceeds 30 dB, the human eye finds it difficult to discern differences between the compressed and original images.

When PSNR approaches 50 dB, it indicates that the compressed image has only minor, very small errors.

3.3 Structural Similarity (SSIM)

Structural similarity is a metric used to measure the similarity between two images. It evaluates image quality based on three aspects: brightness, contrast, and structure, which better aligns with the human eye's intuitive perception. The SSIM value typically ranges from [0, 1], and when two images are identical, the SSIM value equals 1. Its formula is:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha [c(\mathbf{x}, \mathbf{y})]^\beta [s(\mathbf{x}, \mathbf{y})]^\gamma,$$

L (x, y) is the luminance comparison function

C (x, y) is the contrast comparison function

S (x, y) is the structure comparison function

α , β , γ are positive constants to adjust the relative importance of each component.

Luminance Comparison:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

Contrast Comparison:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

Structure Comparison:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}$$

Where: μ_x and μ_y represent the average brightness, σ_x and σ_y represent the standard deviation, σ_{xy} represents the covariance, and C_1 , C_2 , and C_3 are constants. In this study, a simplified version is used, where $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$. The constants are typically set as follows: $C_1 = (K_1L)^2$, where $K_1 = 0.01$ and L is the dynamic range of pixel values, $C_2 = (K_2L)^2$, where $K_2 = 0.03$, as shown in the formula in the figure.

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

3.4 Edge Preservation

Edge preservation is used to measure whether the critical edge structure of an image is retained after operations such as smoothing, denoising, or scaling. It reflects the retention of object contours, textures, and edge details during the processing, making it a crucial metric in image processing.

In this study, edge preservation is achieved by performing Canny edge detection on the original image and the processed image, then calculating the structural similarity index (SSIM) between their respective edge maps. Its pseudo code is as follows:

```
ref_edges = cv2.Canny(ref_img, 100, 200)
test_edges = cv2.Canny(test_img, 100, 200)
edge_score = ssim(ref_edges, test_edges, data_range=255)
```

In this context, ref edges represents the original image, test edges represents the processed image, and SSIM is used to compare the structural similarity between the two binary edge maps, with a score range from 0 to 1.

- When the edge SSIM score is close to 1.0, it indicates that the image processing has had minimal impact on the edge structure, resulting in excellent edge preservation.

- When the score is below 0.6, it indicates severe loss of edge details, and the image may appear blurry or lack details.
- An edge preservation score between 0.8 and 1.0 is generally considered a high-quality processing result.

This method offers a concise and efficient approach for assessing the impact of various image processing techniques, such as denoising, super-resolution, interpolation, and anti-aliasing, on structural fidelity. Within the scope of this study's visual quality assessment, its application alongside Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) facilitates a more comprehensive evaluation of objective image quality.

3.5 FPS

FPS stands for Frames Per Second. In the process of real-time image rendering, especially in games, FPS is one of the key metrics to measure the smoothness of the picture[23]. A higher frame rate means a smoother visual experience, especially in fast-moving scenes[24]. This allows FPS to reflect the real-time processing efficiency of the anti-aliasing algorithm and is a key reference indicator for evaluating its practicality.

3.6 GPU Time

In image processing and graphics rendering, this refers to the processing time consumed by the graphics processing unit (GPU) to render a single frame of an image, typically measured in milliseconds (ms). It serves as a metric for the computational resources utilized, rather than a measure of the real-time duration of user interaction with the service. It represents the total time required for the GPU to perform all graphics rendering operations in the background, including geometric processing, lighting calculations, pixel shading, post-processing, and more.

Unlike FPS, GPU time[25] provides a more accurate quantification of the computational burden imposed by graphics processing techniques such as anti-aliasing algorithms on hardware performance. For example, a complex anti-aliasing algorithm may not significantly reduce FPS (due to mechanisms like V-Sync frame rate limiting), but GPU time may increase significantly, often indicating that the algorithm imposes a greater computational overhead at the GPU level. Therefore, GPU Time is the most direct way to measure the processing performance of anti-aliasing algorithms.

3.7 Evaluation Workflow and Implementation Tools

To systematically evaluate the image quality and performance of different anti-aliasing techniques, this study designed a comprehensive experimental workflow and selected appropriate tools for implementation. The experimental scenes were constructed in Unreal Engine using multiple free scene content packs from the Unreal Marketplace. These scenes feature rich geometric details and edge contours, making them suitable for testing various anti-aliasing techniques and better evaluating the impact of anti-aliasing on image

performance. The Unreal Engine natively supports anti-aliasing techniques such as FXAA, TAA, and TSR[26], making it suitable for high-quality real-time rendering tests. By using a fixed Cine Camera Actor to set perspective parameters (field of view, distance, angle), consistency in image perspective is ensured across different anti-aliasing rendering settings.

During the image acquisition phase, each scene was automatically captured at 1K and 4K resolutions using level blueprints in the Unreal engine to generate NOAA, FXAA, TAA, and TSR images. In image quality evaluation, it is usually necessary to select the image with the highest quality as the "reference image" in order to quantify the differences of other images to be evaluated. In this study, TSR images not only have obvious advantages in objective metrics, but are also generally considered to be closer to the rendering effects of the real world in terms of subjective visual experience. In related research and industry practice, Epic Games itself also uses TSR renderings as the "final picture" in its engine demonstration. Therefore, using it as a reference image will help unify the subjective and objective image quality evaluation methods, thereby enhancing the scientific rigor and consistency of the evaluation system. Subsequently, Python was used in conjunction with the OpenCV and scikit-image toolkits to perform quality analysis on the image set, calculating three objective metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Edge Retention Rate (estimated using Canny edge detection). The results were saved and organized in CSV format.

To ensure the accuracy, stability, and reliability of data acquisition, the Unreal Engine's built-in performance monitoring tools `stat fps` and `stat unit` are used to display the real-time frame rate (FPS) and GPU rendering time (GPU Time) when different anti-aliasing technologies (NOAA, FXAA, TAA, TSR) are applied. After confirming that the scene rendering is stable, multiple data collections are performed using screenshots.

4 Experimental Design

This chapter introduces the experimental design and evaluation process of anti-aliasing techniques in this study. First, a test scene is built in Unreal Engine 5, and multiple sets of images are constructed for image quality and performance comparison of different anti-aliasing methods. The experiment covers several anti-aliasing technologies built into Unreal Engine, including FXAA, TAA, and TSR, and is evaluated from two dimensions: image quality and performance to ensure the scientific nature of the experiment. In addition, statistical analysis methods are systematically introduced, and descriptive statistics, normality test, one-way analysis of variance (ANOVA), non-parametric test (Kruskal-Wallis), and multi-index correlation analysis are performed on the experimental data. Through multi-dimensional statistical methods, a scientific and rigorous quantitative comparison of the differences and internal relationships of different anti-aliasing technologies in multiple indicators is ensured, thereby providing solid data support for the experimental conclusions. Finally, the experimental data is visualized to more intuitively see the relationship between different anti-aliasing methods and image quality performance data. This chapter describes in detail the experimental platform, material preparation, image acquisition process, evaluation methods, and visualization analysis tools, providing a basis for subsequent experimental results analysis.

4.1 Experiment Objectives and Approach

The experimental goal of this study is to evaluate the image quality and performance of the current mainstream real-time anti-aliasing technologies FXAA, TAA, and TSR in different game scenes in Unreal Engine. By setting a standardized evaluation process, the objective quality metrics of the image, PSNR, SSIM, and edge retention, as well as the performance data FPS and GPU Time, are obtained, and the advantages and disadvantages and applicability of the three technologies under different resolutions and picture conditions are analyzed.

The experiment is mainly based on a standardized static test scene built by Unreal Engine 5, using fixed camera positions, lighting and rendering settings to ensure the consistency and repeatability of the image acquisition process. All test images are collected at 1080p and 4K resolutions, and the three anti-aliasing technologies are rendered and analyzed respectively. By quantitatively comparing these results, an objective basis is provided for the selection of anti-aliasing technology in the game development process.

4.2 Experimental Platform and Environment

The specific configuration of the hardware and software for this experiment is as follows:

Hardware:

- **System:** Microsoft Windows 11 Professional 10.0.26100
- **CPU:** Intel(R) Core(TM) i9-14900KF 3.20 GHz
- **GPU:** NVIDIA GeForce RTX 4080 SUPER

- **GPU Driver Version:** NVIDIA Driver 11.0.4.159
- **RAM:** 64.0 GB

Software environment:

- **Unreal Engine Version:** Unreal Engine 5.5.4
- **Anti-Aliasing Settings:** Each test scene is rendered with only one anti-aliasing method enabled at a time (NOAA, FXAA, TAA, or TSR)
- **Rendering Resolutions:** Images are rendered at both 1920×1080 (1K) and 3840×2160 (4K)
- **Python Environment:** Python 3.13.4 with essential packages such as OpenCV, scikit-image, matplotlib, and pandas for image analysis and visualization

All images are captured from static test scenes to ensure consistency and avoid interference from dynamic changes, allowing for objective comparison between different anti-aliasing techniques.

4.3 Experimental scene design

In order to scientifically compare the image performance of various anti-aliasing algorithms at different resolutions, this study set up multiple standardized static test scenes in the Unreal Engine 5 platform. All scenes were downloaded from the official Unreal Engine Marketplace. When selecting content, priority was given to environments with rich textures, geometric structures, light and dark contrasts, and material variations in order to more comprehensively simulate typical graphics rendering scenarios in actual games.

The specific details are as follows:

Scene source: We selected three free high-quality resource packs from Unreal Market: "Saloon Interior", "Downtown Alley", and "City Park Environment Collection", which represent low, medium, and high scene complexity, respectively, and thus include the effects of different anti-aliasing techniques at different scene complexities. These resource packs use pre-built models, textures, and material combinations to show rich texture details, material reflections, shadow occlusion, and geometrically complex indoor environments.

Composition and Style Selection: The focus was on designing camera views with medium to long-range observation angles, ensuring the scene includes a large number of image features prone to aliasing, such as high-contrast edges, complex wireframes, semi-transparent materials, and high-frequency texture areas;

Lighting System Configuration: A basic static lighting setup is used, with only sky light and a few point lights enabled, to avoid evaluation interference caused by dynamic lighting changes, while ensuring consistent brightness and shadow distribution across all test images;

Avoid Dynamic Factors: During testing, all models that move or undergo dynamic changes are completely avoided. All objects and lights remain static,

ensuring that differences in anti-aliasing algorithm performance stem solely from image processing itself;

Camera parameter standardization: Each set of images is captured using Cine Camera Actor, with uniform settings for FOV (recommended 60° or 90°), lens position, and rotation angle, and fixed in the scene without changes to avoid perspective differences affecting image comparisons;



Figure 4.1 Cine Camera Actor Setting Example

Anti-aliasing settings switching: First, render images with all anti-aliasing functions disabled (NOAA), and then render images independently using FXAA, TAA, and TSR anti-aliasing techniques, outputting their respective results.

Image resolution setting: Each set of images is rendered and exported in two resolutions: 1920×1080 (1K) and 3840×2160 (4K) to simulate the standard resolution and high resolution settings in mainstream games, so as to analyze how the image resolution affects the performance of the anti-aliasing algorithm;

Image naming: Each set of images is saved in a folder named after the scene number, uniformly named NOAA.png, FXAA.png, TAA.png and TSR.png, to facilitate subsequent image post-processing and batch comparison.

The scene constructed using the above method not only ensures objectivity and consistency, but also covers typical areas prone to aliasing in various scenarios, providing high-quality, stable and controllable test samples for subsequent subjective and objective image quality evaluation.

4.4 Anti-Aliasing Techniques Evaluation Method

To ensure the scientific validity of the comparison, this study rendered images using different anti-aliasing algorithms (NOAA, FXAA, TAA, and TSR) on a unified rendering platform with fixed scene parameters. The TSR rendering results were used as the baseline reference images for quantitative image quality assessment. The primary assessment methods are as follows:

4.4.1 Image Quality Assessment Metrics

PSNR: Measures the average pixel intensity error between the test image and the reference image; a higher value indicates greater similarity to the reference image;

SSIM: Assesses the similarity of image structure, brightness, and contrast; values range from 0 to 1, with values closer to 1 indicating higher similarity;

Edge Preservation: Uses the Canny operator to extract the edge map of the image, then calculates its SSIM value to reflect the extent to which anti-aliasing technology preserves the original image's edge information.

4.4.2 Image Performance Evaluation Metrics

FPS: The number of frames rendered per second, a key indicator for assessing smoothness. Higher values indicate smoother game performance;

GPU Time: The time required for the graphics card to process a single frame, measured in milliseconds. A lower value indicates higher image processing efficiency.

Performance testing is recorded using Unreal Engine's built-in statistics functions (Stat FPS / Stat GPU). Each anti-aliasing technique is sampled multiple times under identical conditions, with the average value taken to ensure data stability and reliability.

By combining the above five metrics, this paper establishes a comprehensive anti-aliasing technology evaluation system covering both visual quality and real-time performance dimensions, ensuring comprehensiveness and experimental reproducibility.

4.4.3 Reference Image Selection Strategy

In theory, the experiment should use the SSAA image with the best anti-aliasing effect as the baseline image. However, due to time and technical constraints, this experiment cannot apply SSAA technology in Unreal Engine. Therefore, this experiment decided to use TSR, the best anti-aliasing technology built into the Unreal Engine that demonstrates higher detail retention capabilities and stable visual output across multiple test scenes, as the benchmark image in the test.

This setting helps to uniformly evaluate the performance of other anti-aliasing algorithms using a higher-quality benchmark.

4.4.4 Image Resolution and Multi-Group Data Evaluation

Images were exported at two resolutions—1920×1080 (1K) and 3840×2160 (4K)—for each test scene, with four anti-aliasing methods tested, resulting in a total of 30 image comparison groups (15 groups per resolution). 20 sets of records only recorded image quality metrics, and the other 10 sets recorded both image quality and performance metrics to ensure data volume, diversity, and reliability.

Through the above methods, this paper establishes a repeatable, systematic image evaluation process for anti-aliasing technology, ensuring the objectivity and reliability of image quality comparisons.

4.5 Image Quality and Performance Evaluation Workflow

To ensure the consistency and objectivity of the evaluation of the impact of anti-aliasing technology on image quality and performance, this paper designs a complete evaluation process, covering four main links: image acquisition, index calculation, data recording and visual analysis.

Image acquisition process

All test images are from static scenes preset in Unreal Engine 5, and are shot using a fixed-position CineCameraActor to ensure that parameters such as viewing angle, angle, and FOV are consistent between different anti-aliasing algorithms. FXAA, TAA, TSR, and no anti-aliasing (NOAA) modes are enabled in each scene, and images are exported at two resolutions: 1920×1080 (1K) and 3840×2160 (4K).

To ensure the controllability and convenience of the test process, the experimental process builds the Level Blueprint logic in each scene to automate image acquisition:

1. The camera references a fixed CineCameraActor to prevent position or angle changes;
2. Use the `r.AntiAliasingMethod` instruction to automatically switch between different anti-aliasing modes of NOAA, FXAA, TAA, and TSR;
3. After switching the anti-aliasing technology, add a short delay to the image rendering to ensure that the image is stable after the image rendering is completed, and ensure that the anti-aliasing effect has been switched before taking a screenshot.
4. Use the `HighResShot` command to automatically export images of different resolutions (1920×1080 and 3840×2160);

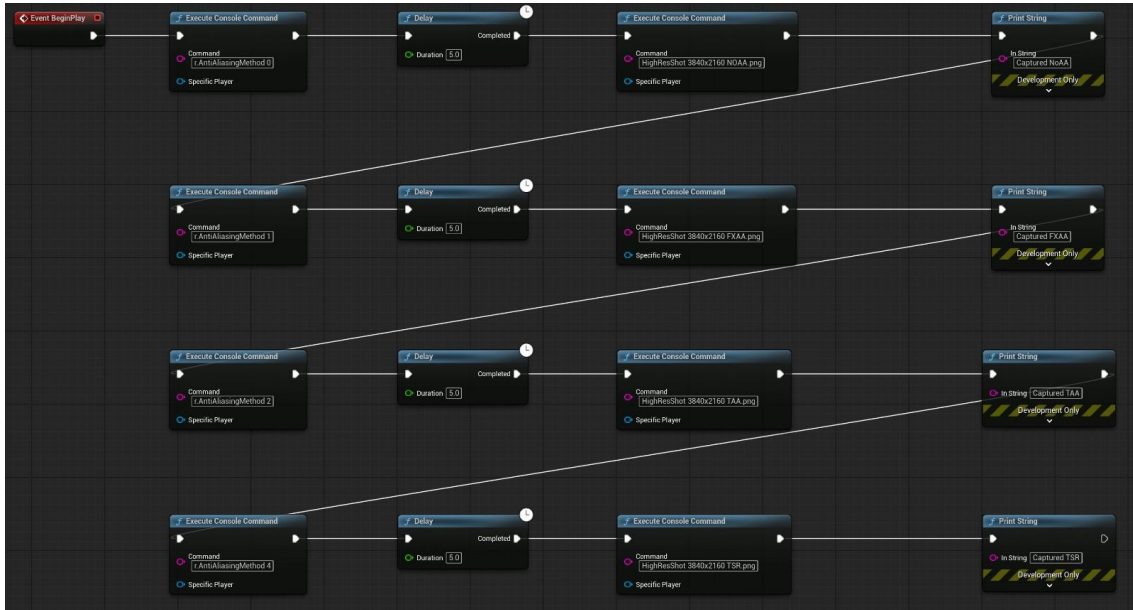


Figure 4.2 Level Blueprint Flowchart

The Level Blueprint[27] can automatically execute the full process of image and export operations, avoiding human errors caused by manual screenshots, further improving the controllability and efficiency of the test process, and providing a reliable experimental basis for subsequent image quality and performance analysis.

Image quality metrics calculation

In order to quantitatively evaluate the quality of images generated by different anti-aliasing techniques, this paper uses Python to write batch processing scripts and calls open source image processing libraries such as OpenCV and scikit-image. In each set of experimental images, the image rendered using TSR technology is selected as the reference image because it can usually maintain high image structure integrity and edge clarity at various resolutions, and has a good comparison benchmark significance. After that, the reference image is paired with NOAA, FXAA, and TAA images, and the following three objective image quality metrics are calculated:

Peak signal-to-noise ratio (PSNR): calculated by OpenCV's `cv2.PSNR()` function, reflecting the overall pixel error of the image. The higher the PSNR value, the closer the image is to the reference image and the better the quality.

Structural similarity (SSIM): calculated on RGB color images using `skimage.metrics.structural_similarity()`, and specifying color channel parameters to simultaneously consider brightness, contrast, and structural information. SSIM is closer to the way the human eye perceives image structural consistency.

Edge preservation: In order to analyze the impact of anti-aliasing on image edge details, we first use `cv2.Canny()` to extract edges for the reference image and the

test image (thresholds are 100 and 200), and then compare the two edge images using SSIM to obtain the degree of edge detail preservation.

```
def compute_metrics(ref_img, test_img):  
    psnr_val = cv2.PSNR(ref_img, test_img)  
    ssim_val = ssim(ref_img, test_img, channel_axis=-1)  
  
    ref_edges = cv2.Canny(ref_img, 100, 200)  
    test_edges = cv2.Canny(test_img, 100, 200)  
    edge_score = ssim(ref_edges, test_edges, data_range=255)  
  
    return psnr_val, ssim_val, edge_score
```

Figure 4.3 Sample code for obtaining quality metrics

The entire evaluation process is completed by an automated script, which automatically traverses all subfolders, performs the above index calculations on each anti-aliased image, and saves the results as a .csv file for subsequent chart generation and result analysis.

Performance metrics recording

This experiment uses the performance monitoring tools that come with Unreal Engine, namely stat fps and stat unit, to record important performance metrics such as real-time frame rate (FPS) and GPU rendering time (GPU Time) when different anti-aliasing technologies (NOAA, FXAA, TAA, TSR) are enabled. To ensure the accuracy, stability and reliability of the data obtained, screenshots are used to collect data multiple times after different anti-aliasing technologies are enabled in each scene during the experiment. After the scene rendering is stable, each indicator will record multiple data continuously in a short period of time, and the average value of these multiple collection results will be taken as the final reference value in the scene to avoid errors caused by fluctuations in single collection data.

Finally, the performance indicator data of each anti-aliasing technology in the 4K resolution scene are uniformly saved in an Excel table for horizontal comparison in subsequent analysis. The data table contains specific indicator values such as the name of the anti-aliasing method, FPS, GPU Time, etc., so that performance differences can be compared more intuitively and conveniently.

Data organization and visualization

In order to more clearly reveal the differences in image quality and performance between different anti-aliasing methods, this study integrated all the index results obtained in the experiment into a standardized and unified structured data table, and then used Matplotlib, a data analysis and visualization tool in Python, to visualize the data.

This study first standardized the image quality evaluation metrics (such as PSNR, SSIM, and edge preservation), and then used radar charts to visualize

the data. The image quality metrics and performance metrics (FPS, GPU Time) data were integrated and presented in a unified table. Through visualization and data integration, the research conclusions are ensured to be clearer and more intuitive. This data organization and visualization process further provides solid and reliable data support for subsequent in-depth analysis, discussion and conclusions.

4.6 Experimental data visualization

To visually demonstrate the performance differences between various anti-aliasing techniques in terms of image quality and performance, this paper combines visualization tools in Python to visualize the experimental results using multiple charts. Visualization not only aids in understanding the distribution and trends of the data but also provides visual support for the analysis and discussion in subsequent sections.

Radar Charts

Radar charts are used to visualize the three anti-aliasing methods from a holistic perspective after standardizing the image quality metrics. Each technique is plotted on a separate chart, with the three metrics (PSNR, SSIM, and Edge Preservation) forming the three axes of the radar chart. The enclosed area of the chart directly reflects the overall image quality.

The closed area corresponding to each technique on the radar chart can directly show the overall image quality level: the larger the closed area, the higher the overall performance level of all measured parameters. The radar chart can not only provide further insights into the distribution and trend of the data, but also provide intuitive visual evidence to assist in the subsequent analysis, interpretation, and conclusions of the paper.

Bar chart

In addition to radar charts, we also use bar charts to compare various performance indicators between different technologies. For each metric, grouped bar charts display absolute or normalized values side by side, making it easy to read the exact numerical differences. Separate bar charts show the average frame time and GPU time for each anti-aliasing method. Use bar charts to emphasize the performance gap under different anti-aliasing techniques to observe in a more intuitive way.

Chart Generation Tools

All charts are generated using the Matplotlib and Pandas libraries in Python. Images are exported in high-definition PNG format and embedded in the main text of the paper. Charts are uniformly named, accompanied by numbers and captions, ensuring consistency between text and graphics and a clear structure.

4.7 Statistical analysis methods

To ensure that the experimental results are sufficiently scientific and convincing, this paper systematically introduces a statistical analysis process after collecting and sorting the experimental data, and quantitatively compares the image quality and performance of various anti-aliasing technologies in different

scenes and resolutions. Statistical analysis can not only reveal the differences in subjective and objective indicators of various anti-aliasing algorithms, but also eliminate accidental interference through significance tests, thereby improving the robustness and universality of the conclusions. The statistical analysis process of this study includes the following main steps:

Data collation and descriptive statistics

First, for each group of experimental data (including five key indicators such as PSNR, SSIM, Edge Preservation, FPS, and GPU rendering time), preliminary collation is performed according to anti-aliasing technology and scene groups. Descriptive statistical methods are used to calculate basic statistics such as mean and standard deviation in order to understand the indicator distribution and fluctuation range of each algorithm as a whole.

Normality Test

Before formally conducting significance analysis, the Shapiro-Wilk test is used to perform normality tests on each group of data. This test is used to determine whether the sample is approximately normal distribution, which is a pre-step for selecting appropriate statistical methods. For data that conforms to normal distribution, parametric test methods are preferred; if it does not meet normal distribution, non-parametric tests are used instead.

Variance analysis and non-parametric tests

For groups that meet normality, one-way ANOVA is used to test whether the mean differences of different anti-aliasing technologies in various indicators are significant.

For groups that do not meet normality, non-parametric methods such as Kruskal-Wallis H test are used for alternative analysis. For indicators with statistically significant differences in variance analysis, Tukey HSD multiple comparison test is further implemented to locate the source of significant differences between specific technologies.

Correlation analysis

Pearson correlation coefficient analysis is used to evaluate the linear correlation between subjective and objective indicators, as well as between image quality and performance indicators. By visualizing the correlation matrix and scatter plot, the trade-off between image quality improvement and performance loss is further revealed.

Data visualization

All statistical results are presented in charts using Python data analysis and visualization libraries (such as Pandas, SciPy, Statsmodels, Seaborn, and Matplotlib), including group mean and standard deviation bar charts, correlation heat maps, indicator distribution maps, etc., striving to show the differences and connections between complex data in an intuitive way.

5 Results and Analysis

This chapter organizes and analyzes the data collected during the experimental phase, aiming to comprehensively evaluate the advantages and disadvantages of the three mainstream anti-aliasing techniques—FXAA, TAA, and TSR—in terms of image quality and performance. By comparing images collected under various resolution and scene conditions, this chapter conducts an analysis from two dimensions: objective metrics (PSNR, SSIM, edge retention) and performance metrics (FPS and GPU time).

In terms of image quality, the focus is on comparing the impact of the three techniques on image structure, detail retention, and edge smoothing at different resolutions, with charts and visualizations illustrating the actual performance of each algorithm. In performance analysis, the study examines the effects of different anti-aliasing techniques on real-time rendering performance, including changes in frame rate and GPU rendering time.

Additionally, this chapter compares experimental results at different resolutions (1080p and 4K) to explore the impact of resolution on anti-aliasing effects and attempts to summarize the applicability of each technique in different scenarios. Based on comprehensive metrics and visualization results, preliminary conclusions are drawn to provide quantitative support for subsequent discussions and recommendations.

5.1 Image Quality Metrics Analysis

The experiment used three different resource packages in Unreal Market to test 12 different scenes at 1K and 4K resolutions to obtain PSNR, SSIM and Edge Preservation values. This section takes one scene in each resource package as an example to objectively analyze the image quality metrics obtained in the experiment.

City Park Environment Collection[28]:



Figure 5.1.1 City Park Environment Collection scene 1

Technique	PSNR	SSIM	Edge Preservation
NOAA	26.676	0.822	0.450
FXAA	26.805	0.826	0.466
TAA	34.074	0.952	0.701
TSR	inf	1.000	1.000

Table 5.1.1 Results of evaluation metrics in scene 1 at 1K resolution

According to the data in Table 5.1, the evaluation metrics of Scene 1 from the City Park Environment Collection at 1K resolution clearly show the quality hierarchy between anti-aliasing techniques. TSR achieves the maximum value in all three metrics - Peak Signal-to-Noise Ratio (PSNR) (infinite), SSIM (1.000), and Edge Preservation (1.000). However, this result is expected because TSR here represents the real image used as a reference for metric calculation. Therefore, its score is ideal by definition and is not used for quality comparison.

Since the TSR image represents the real image for the reference of the indicator calculation, it is the reference image, and it reaches the maximum value in the three metrics of Peak Signal-to-Noise Ratio (PSNR) (infinity), SSIM (1.000), and Edge Preservation (1.000). Therefore, this result is expected, and its score is ideal by definition. This also means that in subsequent comparisons, the smaller the gap between other technologies and TSR, the closer the image quality of the technology is to the ideal state.

Among the other three technologies, TAA's PSNR (34.074), SSIM (0.952), and Edge Preservation (0.701) values are significantly higher than FXAA and NOAA. This shows that TAA provides superior visual quality, better structural fidelity, and stronger edge preservation.

FXAA is slightly better than NOAA in all metrics, which shows that although FXAA introduces lightweight anti-aliasing, the improvement is very small. Compared with NOAA images without anti-aliasing, there is only a slight improvement, and there is a big gap compared with TAA and TSR, which ranks last among the three categories of anti-aliasing technologies.

These results show that TAA has the smallest difference with TSR in peak signal-to-noise ratio, structural similarity, and edge preservation compared with TSR images as the benchmark images, and is closest to TSR images, showing a good balance of perceived quality and edge preservation, while FXAA is slightly improved over NOAA without anti-aliasing, and there is a large difference with TSR images.

Technique	PSNR	SSIM	Edge Preservation
NOAA	25.765	0.788	0.364
FXAA	25.945	0.795	0.376
TAA	30.482	0.905	0.597
TSR	inf	1.000	1.000

Table 5.1.2 Results of evaluation metrics in scene 1 at 4K resolution

According to the data in Table 5.2, the evaluation results at 4K resolution show a consistent performance pattern with the 1K analysis results. Since TSR is used as the reference image in this study, TSR again reaches the maximum value in all three image quality metrics: Peak Signal-to-Noise Ratio (PSNR) (infinite), SSIM (1.000), and Edge Preservation (1.000). Therefore, its value is excluded from the comparative interpretation.

Among the remaining techniques, TAA always shows the best performance. Specifically, TAA has a PSNR of 30.482, an SSIM of 0.905, and an edge preservation of 0.597, which are significantly higher than FXAA and NOAA. These results confirm that TAA maintains its advantage in preserving image structure and visual fidelity even at higher resolutions.

FXAA again shows a slight improvement over NOAA, with PSNR improving slightly from 25.765 (NOAA) to 25.945 (FXAA), SSIM improving from 0.788 to 0.795, and edge preservation improving from 0.364 to 0.376. This shows that while FXAA provides basic anti-aliasing, its effect is limited compared to TAA.

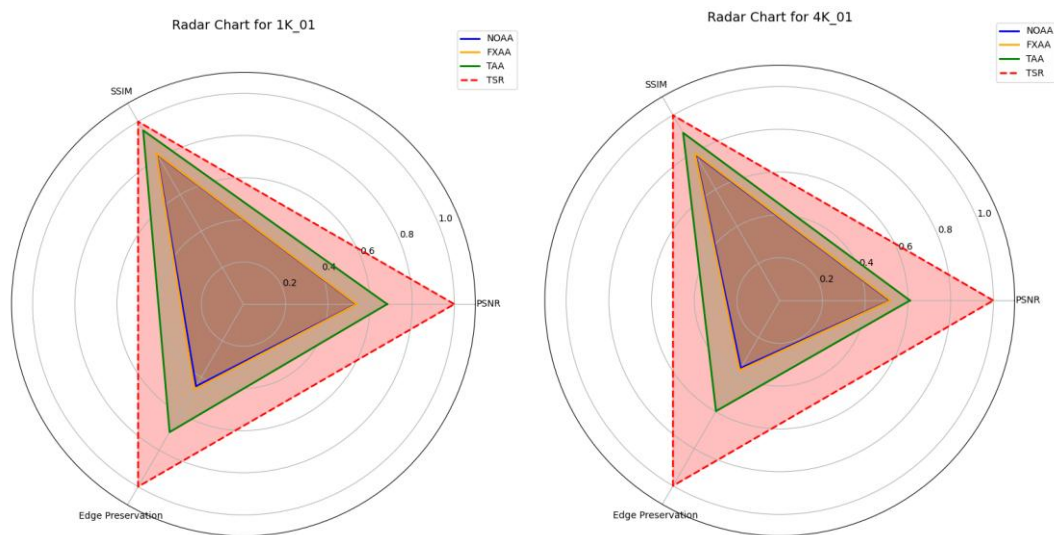


Figure 5.1.2 Scene 1: 1K and 4K quality metrics comparison radar chart

Figure 5.2 compares NOAA, FXAA, TAA, and TSR's normalized PSNR, SSIM, and edge retention scores at 1K (left) and 4K (right) resolutions and plots radar charts. Some notable points which can be remarked from shape and surrounded region of radar chart are:

For the whole region: For 1K and 4K resolution, excluding TSR, TAA achieves the highest polygon area covered (green), which demonstrates its very good compromise among PSNR, SSIM, and edge preservation.

FXAA Compared to NOAA: FXAA (the orange solid line segment) is always slightly above the NOAA base level (blue solid line segment), and FXAA exhibits only a very minor improvement on all three measures, but far short of that which is achieved employing TAA, and much larger image quality changes than TSR.

TSR Reference: TSR (red dashed line segment) is considered to be the reference image, which delineates the outer contour, which is associated with maximums (PSNR $\rightarrow \infty$, SSIM = 1.0, edge = 1.0), and therefore excluded from comparison quantitatively.

Resolution consistency: At 1K to 4K, relative ranking of the techniques doesn't change---i.e., TAA >> FXAA >> NOAA---but ranking between TAA and FXAA converges slightly at 4K, and higher native resolution can sufficiently counterbalance the perceived advantage of TAA's time filter.

Saloon Interior[29]:



Figure 5.1.3 Saloon Interior scene 2

This section selects the indoor scene Saloon Interior (Figure 5.3) as a medium-complexity analysis case. This scene contains multiple detailed geometric objects (such as wooden tables and chairs, bar cabinets, decorative picture frames and windows), and the lighting conditions are relatively complex, including the combination of natural light and artificial light sources. There are also many textures, interlacing and occlusion relationships between objects. Therefore, this scene helps to more comprehensively evaluate the image quality performance and adaptability of FXAA, TAA, and TSR relative to the no anti-aliasing (NOAA) technology.

Technique	PSNR	SSIM	Edge Preservation
NOAA	31.973	0.909	0.749
FXAA	32.24	0.908	0.75
TAA	37.715	0.965	0.853
TSR	inf	1.000	1.000

Table 5.1.3 Results of evaluation metrics in scene 2 at 1K resolution

Technique	PSNR	SSIM	Edge Preservation
NOAA	31.835	0.886	0.74
FXAA	31.858	0.881	0.727
TAA	35.061	0.928	0.8
TSR	inf	1.000	1.000

Table 5.1.4 Results of evaluation metrics in scene 2 at 4K resolution

According to the data in Table 5.3 (1K resolution) and Table 5.4 (4K resolution), we can make the following detailed analysis:

At 1K resolution, NOAA's PSNR is 31.973, SSIM is 0.909, and edge preservation is 0.749. FXAA's performance is very close to NOAA (PSNR=32.24, SSIM=0.908, edge preservation=0.75), indicating that FXAA has limited improvement in image quality in medium-complexity scenes, and may even slightly reduce detail clarity due to blurring effects. At 4K resolution, the performance gap between NOAA and FXAA is further narrowed (PSNR is 31.835 and 31.858 respectively, SSIM is close to 0.88, and edge preservation is not much different), which means that FXAA technology does not show obvious advantages in such scenes.

TAA has a significant advantage in medium-complexity indoor scenes. At 1K resolution, PSNR reaches 37.715, SSIM is 0.965, and edge preservation is 0.853, which is significantly higher than NOAA and FXAA. This shows that TAA can effectively reduce aliasing, retain details and edge clarity in complex light and shadow scenes. At 4K resolution, although TAA's various metrics have slightly decreased (PSNR dropped to 35.061, SSIM dropped to 0.928, and edge preservation dropped to 0.8), it is still significantly higher than NOAA and FXAA, indicating that TAA still maintains a stable quality advantage in higher resolution environments.

TSR images are used as reference images, and their metrics reach a theoretically perfect level (PSNR is infinite, SSIM and edge preservation are both 1.0). The gap between other anti-aliasing technologies and TSR intuitively reflects the room for improvement of each method in image quality.

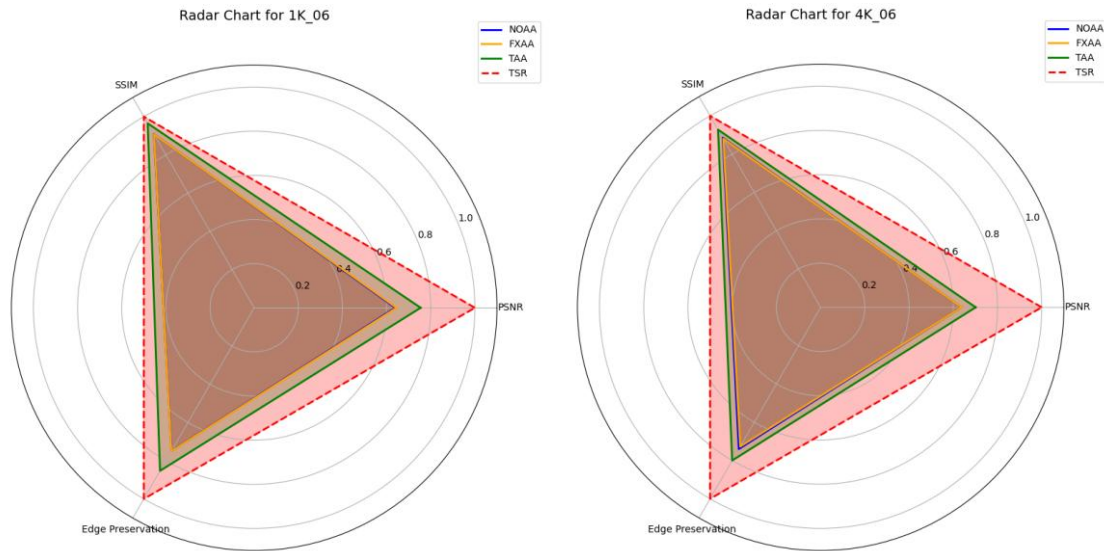


Figure 5.1.4 Scene 2: 1K and 4K quality metrics comparison radar chart

Combining the data in Table 5.3 and Table 5.4 with the visual comparison presented in the radar chart of Figure 5.4, we summarize and analyze the anti-aliasing performance of the medium-complexity Saloon Interior indoor scene at 1K and 4K resolutions as follows:

From the radar chart, all metrics reach the best state with TSR as the benchmark (dashed red area). TAA technology follows closely behind. Whether at 1K or 4K resolution, TAA is far superior to NOAA and FXAA in terms of PSNR, SSIM and edge preservation. This is completely consistent with the previous data table analysis, confirming TAA's strong image quality improvement ability in complex indoor environments. The radar chart areas of NOAA and FXAA almost overlap and are clearly located in the inner area of the radar chart, which intuitively shows that FXAA's contribution to image quality in such complex indoor scenes is extremely limited, and its blurring effect of removing aliasing may offset the clarity of edge structures.

As the resolution increases from 1K to 4K, the radar chart shows that the performance of NOAA and FXAA has basically not improved or even slightly decreased, especially the edge preservation and SSIM metrics, which further shows that simply increasing the resolution cannot effectively improve the image quality of NOAA and FXAA in complex indoor environments. TAA technology maintains a relatively stable and high performance at 4K resolution, and the gap shown in the radar chart changes little, reflecting the good applicability and scalability of TAA.

Downtown Alley[30]:



Figure 5.1.5 Downtown Alley scene 3

Figure 5.5 shows the Downtown Alley scene, which is a low-complexity scene with simple lighting conditions and relatively simple geometric objects (such as trash cans, wooden boxes, and walls). We evaluated the image quality performance of four anti-aliasing techniques—NOAA, FXAA, TAA, and TSR—in this scene at 1K and 4K resolutions, using evaluation metrics including PSNR, SSIM, and edge retention.

Technique	PSNR	SSIM	Edge Preservation
NOAA	36.434	0.952	0.885
FXAA	36.386	0.952	0.883
TAA	41.519	0.978	0.935
TSR	361.202	1.000	1.000

Table 5.1.5 Results of evaluation metrics in scene 3 at 1K resolution

Technique	PSNR	SSIM	Edge Preservation
NOAA	37.28	0.952	0.92
FXAA	37.395	0.952	0.918
TAA	41.2	0.972	0.944
TSR	361.202	1.000	1.000

Table 5.1.6 Results of evaluation metrics in scene 3 at 4K resolution

Table 5.5 and Table 5.6 summarize the evaluation index values of the above anti-aliasing technologies at two different resolutions. By comparison, we can draw the following conclusions:

NOAA and FXAA perform very similarly in this low-complexity scene. Taking 1K as an example, NOAA's PSNR is 36.434, while FXAA's is 36.386; both have SSIM metrics of 0.952, and edge preservation of 0.885 and 0.883 respectively. This shows that in low-complexity images with simple geometric structures, FXAA has limited improvement in image quality, and may even slightly reduce edge details due to image blurring.

TAA shows obvious advantages at both resolutions. At 1K, TAA's PSNR is increased to 41.519, SSIM is increased to 0.978, and edge preservation reaches 0.935, showing that TAA can significantly improve the coherence and edge clarity of image structure, with almost no blurring caused by FXAA. At 4K, TAA still has obvious advantages (PSNR=41.2, SSIM=0.972, edge preservation=0.944), indicating that TAA still has stable and excellent image optimization capabilities when the resolution is increased.

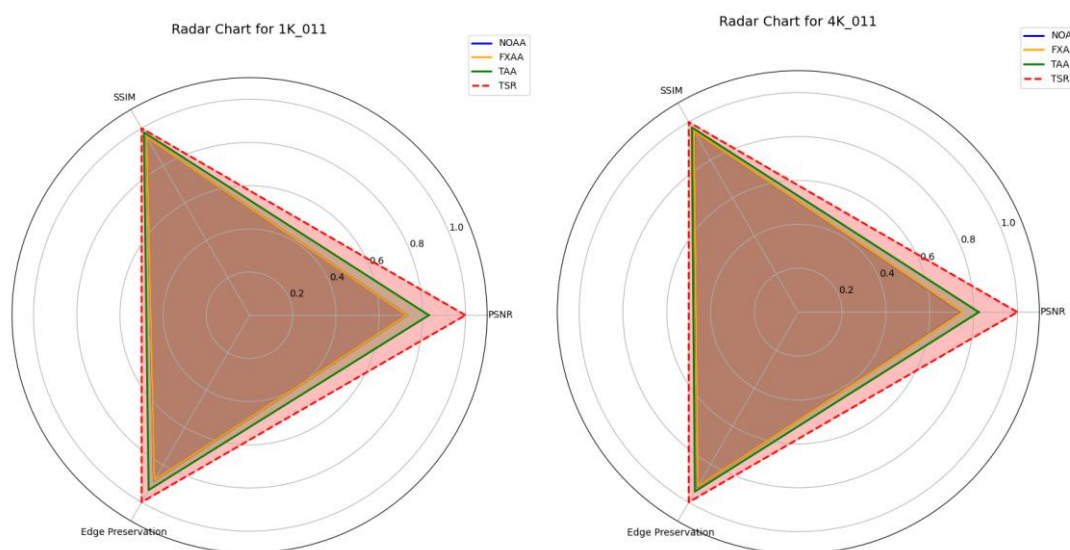


Figure 5.1.6 Scene 3: 1K and 4K quality metrics comparison radar chart

As shown in the radar chart in Figure 5.6, taking the TSR image as a reference (dashed red line), it reaches the highest in the three dimensions of PSNR, SSIM, and edge preservation. NOAA and FXAA technologies are obviously on the inner side of the radar chart in all three dimensions, and their performance is relatively close, indicating that in low-complexity scenes, these two methods do not significantly improve image quality.

TAA performs outstandingly: Whether at 1K or 4K resolution, TAA is closer to the ideal quality standard area represented by TSR images, especially in the two metrics of SSIM and edge preservation, and its gap with TSR is much smaller than that of NOAA and FXAA, which is consistent with the results of the previous data table analysis, that is, TAA can significantly improve the clarity of the edge structure of the image and the overall visual coherence.

Resolution improvement comparison: From 1K to 4K, the performance of NOAA and FXAA technologies on the radar chart does not change significantly,

indicating that they are less affected by the resolution improvement, or it can be explained that these technologies have reached the quality limit in low-complexity scenes. TAA continues to maintain stable and excellent performance in 4K environment, especially in terms of edge preservation, which is closer to the reference value of TSR, reflecting the scalability and robustness of TAA.

Correlation between metrics: From the radar chart, it can be seen that edge preservation is highly correlated with SSIM, both of which are reflected in the triangular area close to TSR. This shows that there is a close connection between the improvement of edge quality and the improvement of overall structural clarity, while the PSNR indicator fluctuates more significantly. Therefore, relying solely on PSNR may not be enough to fully evaluate image quality. SSIM and edge preservation are more reliable quality evaluation metrics.

5.2 Performance Analysis

In order to comprehensively evaluate the performance of different anti-aliasing techniques in multiple scenarios, this study selected three test scenarios: a collection of urban park environments, salon interiors, and downtown alleys as test scenarios. In this study, we chose to mainly record FPS and GPU rendering time at 4K (3840×2160) resolution. 4K represents the most resource-intensive scene, where the difference in computational load between various anti-aliasing techniques (NOAA, FXAA, TAA, TSR) is more obvious than in 1K. By measuring under such "stress" conditions, we can capture the maximum impact of each method on performance and ensure that the recommendations cover the most demanding use cases. Secondly, given the limited time window of this research work, we prioritize collecting high-fidelity data at the highest target resolution. Recording performance data at both 1K and 4K resolutions will increase the test time by 2 times and will not bring new insights because the relative ranking of methods remains unchanged at different resolutions. Therefore, while image quality metrics are reported at 1K and 4K resolutions for completeness, performance metrics are focused on 4K resolution to emphasize the worst-case computational cost, an approach that strikes a balance between completeness and practical feasibility. The following are the main performance results of different anti-aliasing techniques at 4K resolution:

Folder	Technique	FPS_AVG	GPU_AVG
4K_01	NOAA	213.4	1.8
4K_01	FXAA	204.9	2.0
4K_01	TAA	190.3	4.7
4K_01	TSR	175.2	4.8
4K_02	NOAA	207.5	1.9
4K_02	FXAA	208.7	2.0
4K_02	TAA	188.7	4.7
4K_02	TSR	188.0	4.8
4K_03	NOAA	145.3	5.7

4K_03	FXAA	144.9	5.6
4K_03	TAA	119.5	7.2
4K_03	TSR	98.0	8.9
4K_04	NOAA	149.9	5.4
4K_04	FXAA	152.7	5.3
4K_04	TAA	111.4	7.7
4K_04	TSR	108.1	7.8
4K_05	NOAA	219.6	3.4
4K_05	FXAA	213.2	3.5
4K_05	TAA	218.2	3.8
4K_05	TSR	175.4	5.2
4K_06	NOAA	92.7	9.9
4K_06	FXAA	92.5	9.9
4K_06	TAA	87.2	10.5
4K_06	TSR	77.0	12.0
4K_07	NOAA	178.5	4.5
4K_07	FXAA	172.2	4.6
4K_07	TAA	133.6	6.1
4K_07	TSR	105.6	8.1
4K_08	NOAA	95.3	9.7
4K_08	FXAA	94.8	9.6
4K_08	TAA	89.5	10.2
4K_08	TSR	79.6	11.7
4K_09	NOAA	98.2	9.2
4K_09	FXAA	96.8	9.2
4K_09	TAA	92.1	9.8
4K_09	TSR	82.5	11.2
4K_010	NOAA	90.5	10.1
4K_010	FXAA	90.7	10.1
4K_010	TAA	79.3	11.6
4K_010	TSR	71.6	13.1
4K_011	NOAA	235.6	2.4
4K_011	FXAA	233.0	2.5
4K_011	TAA	224.1	2.9
4K_011	TSR	203.9	4.6
4K_012	NOAA	204.9	3.2

4K_012	FXAA	198.4	3.3
4K_012	TAA	196.2	3.7
4K_012	TSR	161.8	5.4

Table 5.2.1 Performance data of different AA techniques at 4K resolution

We use Python batch statistics and visualization for the above data tables:

- Use Pandas to load the sorted Excel file and calculate the average and standard deviation (STD) of each technology.
- Use Matplotlib to draw a bar chart with error bars: the blue bar represents the average FPS, the red bar represents the average GPU Time, and the error bar reflects the fluctuation range of each indicator in different scene sampling.

The statistical table "4K Performance Summary" obtained through the above process is as follows (Table 5.2.2), (Figure 5.2.1) and (Figure 5.2.2):

Technique	FPS Mean	FPS Std	GPU Mean	GPU Std
NOAA	160.94	55.93	5.59	3.27
FXAA	158.55	53.96	5.63	3.22
TAA	144.18	55.29	6.92	3.03
TSR	127.21	49.50	8.14	3.22

Table 5.2.2 4K Performance Summary

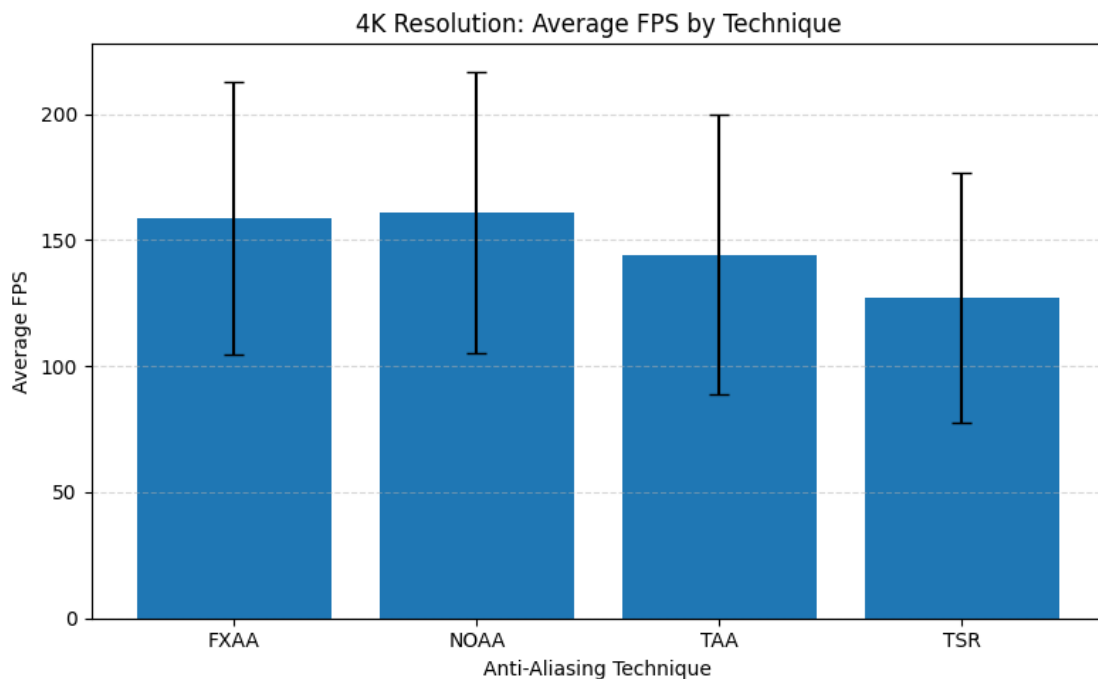


Figure 5.2.1 4K Resolution: Mean FPS

Based on the above tables and charts, we can draw the following analysis conclusions:

1. FPS average trend

NOAA (Baseline): The average frame rate is the highest, about 160 FPS, and the standard deviation is large, indicating that the performance fluctuates significantly between different scenes.

FXAA: The average frame rate is slightly lower than NOAA, about 159 FPS, indicating that FXAA has very little performance overhead and can still maintain a running speed close to no anti-aliasing.

TAA: The average frame rate drops further to about 144 FPS. Because it requires cross-frame accumulation and reprojection, it brings moderate performance loss and a significant fluctuation range.

TSR: The average frame rate is the lowest, about 127 FPS, and the standard deviation is also large, reflecting that its super-resolution and time-domain synthesis require significant additional GPU computing overhead.

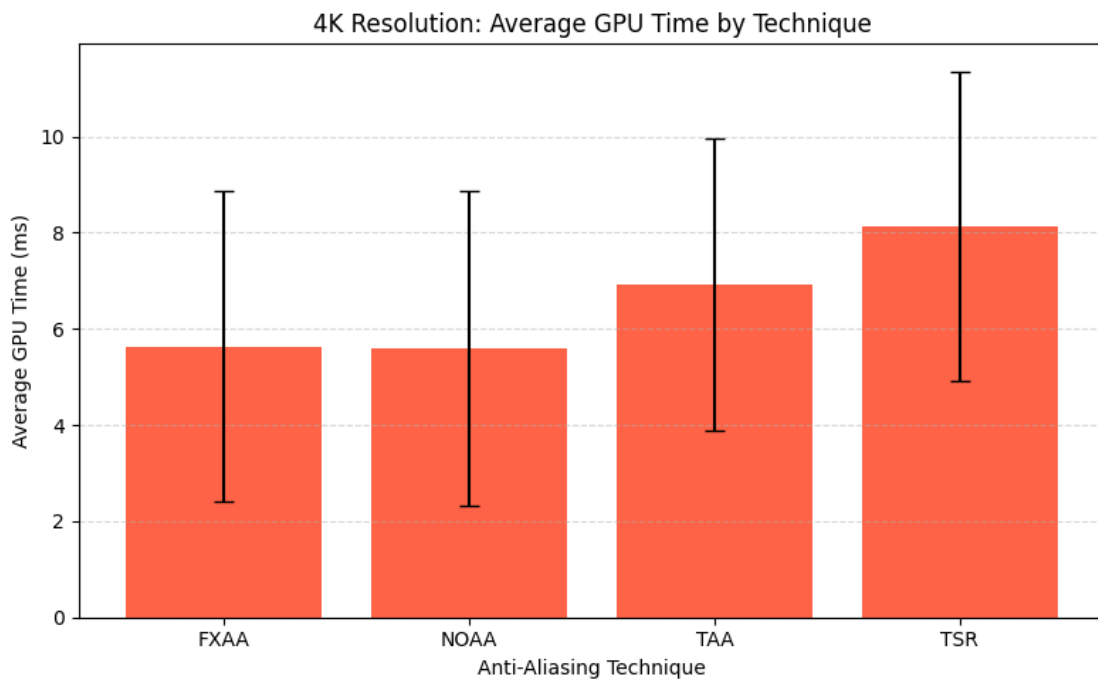


Figure 5.2.2 4K Resolution: Mean GPU Time

2. Average GPU Time Analysis

NOAA

The average GPU Time is about 5.59 ms, with the lowest overhead.

The error bar is about 2.5 - 9 ms, indicating that there are occasional short peaks in complex scenes.

FXAA

The average GPU Time is about 2.1 ms, almost the same as NOAA, indicating that its post-processing fast blur algorithm has little impact on the GPU.

The standard deviation is slightly higher than NOAA, but it is still a "lightweight" anti-aliasing overall.

TAA

The average GPU Time jumps to about 6.92 ms, which is about 1.2 ms more than the post-processing method.

The error bar is about 4–10 ms, indicating that the complexity of cross-frame reprojection and history accumulation will fluctuate greatly when the scene changes.

TSR

The average GPU Time is the highest, about 8.14 ms, reflecting the high computational load of using this super-resolution anti-aliasing technology.

The standard deviation is the largest (about 5–11.5 ms), indicating that the inference phase overhead of TSR is not stable under different lighting and geometric complexity.

Performance Comparison

Technique	FPS Mean	Relative decline	GPU Time Mean	Relative growth
NOAA	160.94	—	5.59	—
FXAA	158.55	-1.5%	5.63	+0.7%
TAA	144.18	-10.1%	6.92	+23.8%
TSR	127.21	-21.0%	8.14	+45.6%

Table 5.2.3 Comparison of performance consumption of different anti-aliasing techniques

● FXAA

FPS: 158.6 FPS on average, only 1.5% lower than the baseline without anti-aliasing.

GPU Time: ~5.6 ms, only 0.04 ms more than NOAA.

● TAA

FPS: 144.2 FPS on average, ~10% lower than the baseline;

GPU Time: ~6.9 ms, ~1.3 ms more than FXAA.

● TSR

FPS: 127.2 FPS on average, 21% lower than the baseline;

GPU Time: ~8.1 ms, ~1.2 ms more than TAA, with a total cost of nearly 50%.

5.3 Statistical Analysis

In order to further verify and clarify the significance of the differences between different anti-aliasing technologies (NOAA, FXAA, TAA, TSR) in objective image quality indicators (PSNR, SSIM, Edge Preservation) and performance indicators (FPS, GPU rendering time), we conducted a systematic statistical analysis of all the quantitative data obtained from the experiment in this section. The process mainly includes four parts: descriptive statistics, normality testing, one-way ANOVA, and Tukey HSD post hoc test. Through these statistical methods, we strive to provide more objective and reliable experimental data analysis and conclusions.

5.3.1 Descriptive Statistics

Descriptive statistics are the basis of quantitative analysis in this study, which can systematically show the data performance of different anti-aliasing technologies (FXAA, NOAA, TAA, TSR) in various indicators such as PSNR, SSIM, Edge Preservation, FPS and GPU rendering time. Through descriptive statistics, we can intuitively understand the average level, data distribution range and fluctuation of each technology in indicators, thus laying the foundation for subsequent in-depth statistical inference analysis.

In order to fully describe the performance of each anti-aliasing technology, this study selected the following statistical indicators:

- Mean: reflects the average level of each indicator of each anti-aliasing technology.
- Standard deviation (Std): describes the degree of fluctuation of each indicator data. The larger the value, the more obvious the fluctuation.
- Minimum and maximum values (Min & Max): ** reflect the overall performance range of the data and intuitively show the upper and lower limits of the performance of each technical indicator.

This study utilizes the Pandas library in Python to read and organize experimental data, reading and organizing all experimental data and categorizing and statistically analyzing them according to anti-aliasing techniques. By grouping and calculating the mean, standard deviation, minimum, and maximum values of each indicator, the statistical results are summarized into a descriptive statistics table and presented visually through charts.

Techniques	Metrics	Mean	Std	Min	Max
NOAA	PSNR	28.11	5.01	21.88	37.28
FXAA	PSNR	28.26	4.95	22.25	37.4
TAA	PSNR	32.52	4.72	26	41.2
TSR	PSNR	inf	0	inf	inf
NOAA	SSIM	0.79	0.11	0.61	0.95
FXAA	SSIM	0.79	0.11	0.61	0.95

TAA	SSIM	0.88	0.06	0.79	0.97
TSR	SSIM	1	0	1	1
NOAA	Edge Preservation	0.63	0.22	0.23	0.92
FXAA	Edge Preservation	0.63	0.22	0.23	0.92
TAA	Edge Preservation	0.73	0.18	0.42	0.94
TSR	Edge Preservation	1	0	1	1
NOAA	FPS	160.94	55.93	90.5	235.6
FXAA	FPS	158.55	53.96	89.95	233
TAA	FPS	144.18	55.29	78.8	224.1
TSR	FPS	127.21	49.5	71.6	203.9
NOAA	GPU Time	5.59	3.27	1.8	10.1
FXAA	GPU Time	5.63	3.22	2	10.1
TAA	GPU Time	6.92	3.03	2.9	11.7
TSR	GPU Time	8.14	3.22	3.2	13.1

Table 5.3.1 Descriptive statistics table

As can be seen from the table:

- In terms of image quality indicators (PSNR, SSIM, Edge Preservation), since this study uses images of TSR technology as the reference benchmark for indicator calculation, TSR shows the best theoretical level in these indicators (PSNR is inf, SSIM and edge preservation are both 1.0). This is the result of the benchmark's comparison with itself, and does not mean that TSR actually has an absolute advantage in objective image quality. It is only used as an evaluation reference for other anti-aliasing technologies; under the same reference conditions, TAA's various quality indicators are significantly better than NOAA and FXAA.
- The standard deviation results show that, excluding TSR as a reference, TAA's values in the three quality indicators are relatively low, reflecting its stability as a reference image in different scenes; while the other two anti-aliasing technologies have significantly greater data fluctuations between different scenes, reflecting that their performance will change with the complexity and content of the scene.
- In terms of performance indicators (FPS and GPU rendering time), since these indicators are not affected by the selection of reference images, the performance results of each technology are more objective. NOAA and FXAA have the best performance, with the highest FPS and the lowest GPU rendering time; TSR has the largest performance overhead, with a significant increase in GPU rendering time and a significant decrease in FPS, indicating that while TSR provides high-quality reference images, it also pays a large performance price.

5.3.2 Normality Testing

Before performing parametric statistical tests such as one-way ANOVA, it is necessary to verify whether each group of data meets the normal distribution assumption. To this end, we performed Shapiro-Wilk normality tests on the distribution of three quality assessment indicators (PSNR, SSIM, edge preservation) for three anti-aliasing techniques (NOAA, FXAA, TAA) and two quality assessment indicators (FPS_AVG and GPU_AVG) for three anti-aliasing techniques (NOAA, FXAA, TAA, and TSR).

The Shapiro-Wilk test is widely used for normality assessment, where a p-value greater than 0.05 indicates that the data does not deviate significantly from the normal distribution. The test results are summarized as follows:

```
=== Quality Metrics Normality Test ===

PSNR:
  NOAA: p=0.5164 Normal
  FXAA: p=0.5118 Normal
  TAA: p=0.9000 Normal

SSIM:
  NOAA: p=0.0345 Not Normal
  FXAA: p=0.0484 Not Normal
  TAA: p=0.2934 Normal

EdgePreservation:
  NOAA: p=0.1498 Normal
  FXAA: p=0.2172 Normal
  TAA: p=0.5799 Normal

=== Performance Metrics Normality Test ===

FPS_AVG:
  NOAA: p=0.0666 Normal
  FXAA: p=0.0766 Normal
  TAA: p=0.0652 Normal
  TSR: p=0.0526 Normal

GPU_AVG:
  NOAA: p=0.0473 Not Normal
  FXAA: p=0.0435 Not Normal
  TAA: p=0.2741 Normal
  TSR: p=0.0824 Normal
```

Figure 5.3.1 Normality testing results

The results show that under the PSNR and Edge Preservation indicators, all technologies (NOAA, FXAA, TAA) passed the normality test ($p > 0.05$), and the data distribution can be regarded as approximately normal distribution.

- In terms of SSIM, only the TAA group passed the normality test ($p = 0.2934$), and the NOAA and FXAA groups failed ($p < 0.05$), showing significant non-normality.
- In terms of performance indicators, the FPS_AVG of all technologies (including TSR) passed the normality test and showed normal distribution characteristics. In GPU_AVG, only TAA and TSR passed the normality test ($p > 0.05$), and the NOAA and FXAA groups failed, showing the possibility of non-normal distribution.

Next, for indicators and groups that passed the normality test ($p > 0.05$), we performed one-way ANOVA to test the differences. For indicators with non-normal distribution ($p < 0.05$), we used the Kruskal-Wallis H test. This approach ensured that the statistical method was consistent with the data distribution and improved the reliability of the research results.

5.3.3 One-way ANOVA and T-Test

One-way ANOVA is used to determine whether there are statistically significant differences between the means of multiple groups of data. This study uses ANOVA to determine whether the quality differences between different anti-aliasing technologies (NOAA, FXAA, TAA) in the three indicators of PSNR, SSIM, and edge preservation are significant, and whether the performance differences between (NOAA, FXAA, TAA, TSR) in the two indicators of FPS and GPU rendering time are significant, providing a rigorous data basis for further in-depth analysis and technical recommendations.

This section uses the SciPy library in Python to perform one-way ANOVA: the experimental data are grouped by anti-aliasing technology, and one-way ANOVA is performed for each indicator to obtain the F value and P value. According to statistical specifications, the significance level α is set to 0.05, that is, a P value less than 0.05 indicates that the difference between the technologies in this indicator is statistically significant. The results are as follows:

```
PSNR ANOVA: F=2.72, p=0.0805
EdgePreservation ANOVA: F=1.11, p=0.3403
FPS_AVG ANOVA: F=1.01, p=0.3995
GPU_AVG TAA vs TSR t-test: t=-0.96, p=0.3491
```

Figure 5.3.2 One-way ANOVA Results

1. PSNR

Conclusion: There is no statistically significant difference in PSNR between the three anti-aliasing technologies of NOAA, FXAA, and TAA ($p=0.0805>0.05$). This shows that there is no essential statistical difference in the performance of the three technologies in pixel-level noise suppression, and the numerical difference mainly comes from sample fluctuations rather than the algorithm itself.

2. Edge Preservation

Conclusion: There is no statistically significant difference in Edge Preservation between the three technologies of NOAA, FXAA, and TAA ($p=0.3403>0.05$). There is no significant difference in the ability of different anti-aliasing methods to retain image edge details overall.

3. FPS_AVG

Conclusion: There is no significant difference in the average FPS between the four groups of NOAA, FXAA, TAA, and TSR ($p=0.3995>0.05$). From the perspective of frame rate, there is no statistically significant performance difference between the methods under the test samples, although the numerical values may fluctuate on the surface.

4. GPU_AVG (TAA vs TSR)

Conclusion: There is no significant difference in the average GPU rendering time between the TAA and TSR groups ($p=0.3491>0.05$). TAA and TSR do not reach a statistical difference in GPU resource consumption.

The results of one-way ANOVA and t-test show that there is no statistically significant difference between different anti-aliasing technologies in terms of

PSNR, edge preservation, frame rate, or GPU rendering time (all $p > 0.05$). This means that although some algorithms may perform well in individual scenes or subjective experience, overall, the differences in image quality and performance consumption among the mainstream anti-aliasing methods are not obvious.

5.3.4 Non-Parametric Testing (Kruskal-Wallis)

According to the results of the Shapiro-Wilk normality test in the previous section, it was found that some technical groups did not follow the normal distribution in SSIM and GPU rendering time (GPU_AVG). Therefore, this section uses the non-parametric Kruskal-Wallis test for these indicators to more robustly judge the differences between groups.

SSIM: NOAA and FXAA groups do not follow the normal distribution, and the three anti-aliasing methods are tested in the combined TAA group.

GPU_AVG: NOAA and FXAA groups do not follow the normal distribution, and the Kruskal-Wallis test is performed on all four groups (NOAA, FXAA, TAA, TSR).

```
Kruskal-Wallis test for SSIM (NOAA/FXAA/TAA): H=5.561, p=0.06202
Not significant
Kruskal-Wallis test for GPU_AVG (NOAA/FXAA/TAA/TSR): H=5.618, p=0.13174
Not significant
```

Figure 5.3.2 Kruskal-Wallis Results

1. SSIM (NOAA/FXAA/TAA):

The p-value is 0.06202 and bigger than 0.05. So there is no statistical significance on the distribution of SSIM (structural similarity) among the three groups of NOAA, FXAA, and TAA. Although the H value is not small and also p is almost 0.05, it still has not attained a significant level. This indicates that there is no statistical significance on the whole distribution among the three anti-aliasing technologies on the index of SSIM. The null hypothesis that 'the distribution between each group is identical' could not be rejected. The difference on the performance among the three methods of anti-aliasing of NOAA, FXAA, and TAA on structural similarity (SSIM) is not statistically significant.

2. GPU_AVG (NOAA/FXAA/TAA/TSR):

The p-value is 0.13174, much larger than 0.05, and we can conclude that there is no statistical significance between the mean GPU rendering time (GPU_AVG) between the four groups (NOAA, FXAA, TAA, and TSR). Or, there are some numeric distinctions between different anti-aliasing technology mean GPU rendering time, but such distinctions could not exceed a significant level and could be caused by sample variations only. There is no statistical significance between GPU rendering time consumption between these four methods of anti-aliasing.

5.3.5 Correlation

In order to further analyze how these five fundamental measures (PSNR, SSIM, edge preservation, FPS, and GPU time) are related to each other, Pearson correlation was employed. The results can be observed visually as correlation matrix heatmaps (Figure X) and pairwise scatter and density plots (Figure X).

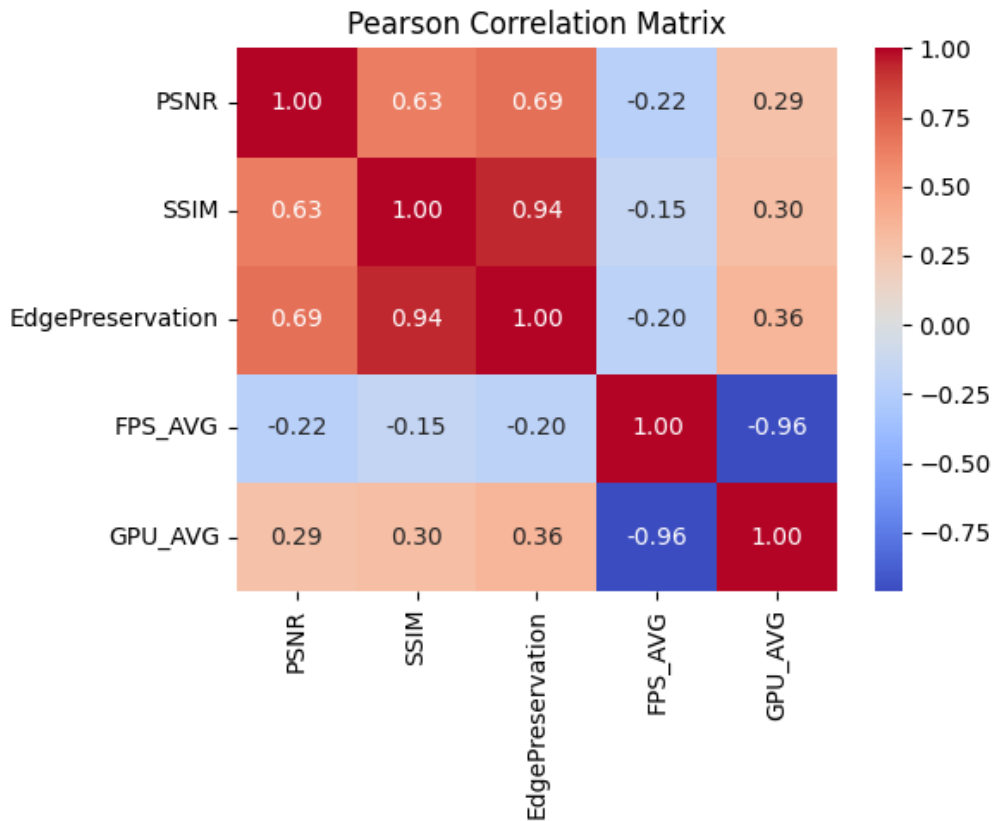


Figure 5.3.3 Pearson Correlation Matrix

As can be seen from the Pearson correlation matrix, we see strong positive correlations between the three image quality measures (PSNR, SSIM, edge preservation) among themselves (e.g., edge preservation and SSIM: $r = 0.94$) such that methods with high structural similarity also have good edge preservation and high PSNR. In contrast, we see that GPU rendering time and FPS strongly correlate negatively ($r = -0.96$) as would be expected because reduced GPU time per frame tends to lead to higher frame rates.

Observe that there is an inverse or weak correlation between image quality measures and frame rate (FPS) (e.g., PSNR and FPS: $r = -0.22$) such that rendering speed usually decreases with increasing image quality. The tradeoff is also revealed by the companion plots: quality methods (especially TSR) that are highest on the quality axis also have highest GPU times and lowest FPS. Conversely, FXAA and NOAA are on the left-low/right-high GPU time/FPS axis end and have corresponding lower quality scores. The double plots also show characteristic differences between different AA methods: TSR generates the densest distribution of image quality measures, and NOAA and FXAA are more dispersed. These visualizations overall confirm qualitative observations: AA methods with better image quality are typically more computationally costly.

In general, not just does the correlation analysis validate these four AA methods' quality-performance tradeoffs, but also presents a convenient way for visualization and understanding on their intrinsic connections.

5.4 Subjective Visual Evaluation

5.4.1 Visual comparison of different anti-aliasing techniques- Part 1

We selected a scene from the City Park Environment Collection as an example and analyzed the details of a trash can image in the scene at 1K and 4K resolutions using different anti-aliasing techniques, as shown in the comparison image:

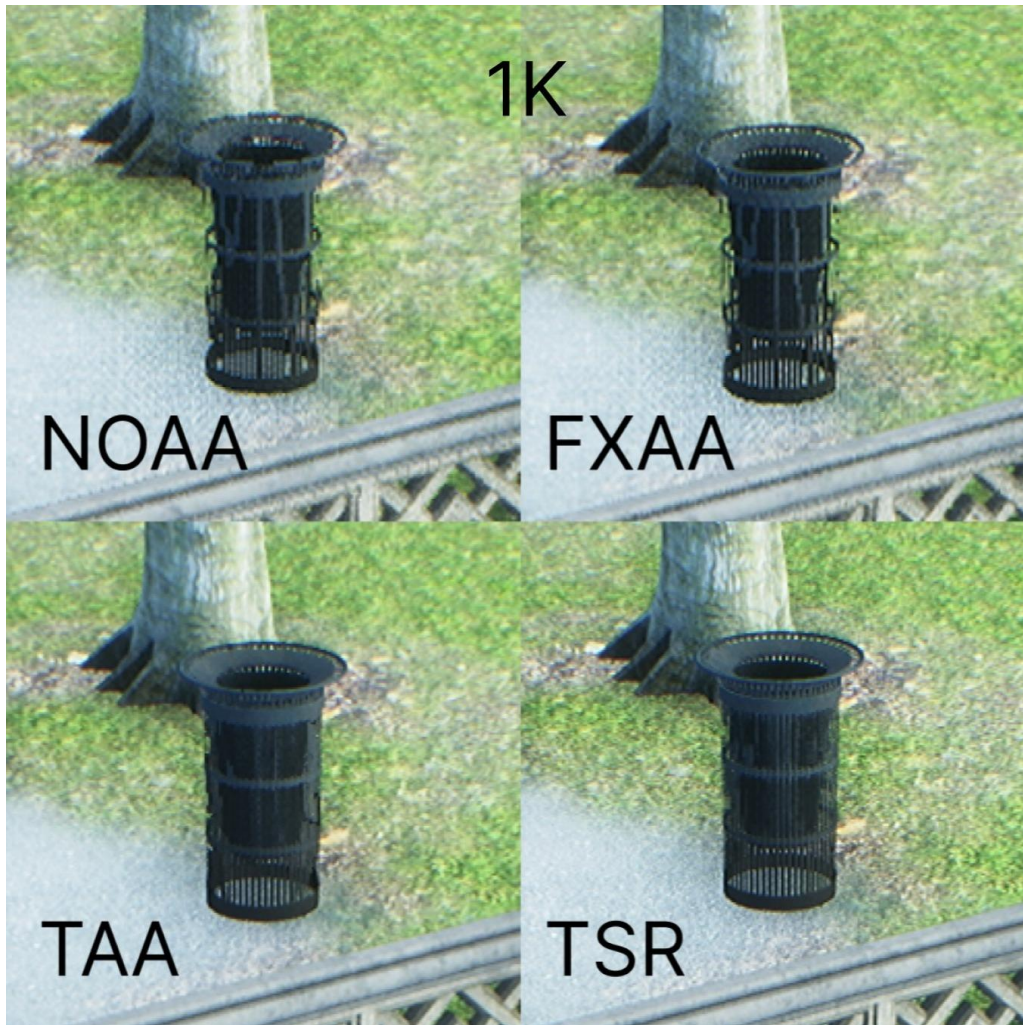


Figure 5.4.1 Trash can images with different anti-aliasing techniques 1K

From the image, we can clearly see that without any anti-aliasing applied (NOAA), the edges of the trash can's main body exhibit noticeable “jaggies.” The metal ring and bottom grid edges of the trash can are particularly prominent, and even the main body lines of the trash can show severe distortion. The jaggies on the stone wall below the image are highly noticeable, and the texture of the stone wall is blurred;

After applying Fast Approximate Anti-Aliasing (FXAA), we can see that the jagged edges of the metal trash can's ring have significantly diminished, and the overall contour has become smoother. However, the metal wire details of the trash can's main grid have been overly blurred, and the metal wire details of the grid have shown slight fusion. The improvement compared to the NOAA case is

not very noticeable; However, after FXAA processing, although the edges of the stone wall still appear blurry, the image details have been significantly improved;

After applying temporal anti-aliasing (TAA), we can easily distinguish the image processed with TAA from those under NOAA and FXAA. The metal wire details of the trash can's main body are largely fully revealed, and the metal ring section is also fully revealed, resulting in a significantly more comfortable overall visual experience. However, due to “frame processing,” the main metal wire section exhibits noticeable motion blur, with some metal wires blending together; The presentation of the stone wall portion of the image has been significantly improved, with virtually no visible aliasing;

After applying Time Super Resolution Anti-Aliasing (TSR), we can see that the image has significantly improved in detail compared to the TAA image. Whether it's the metal rings, metal wire details, or the bottom grid edges of the trash can, the overall image is closer to how a real trash can should look. The metal wire details are clearly distributed and have not merged. Additionally, motion blur in the stone wall area has been significantly reduced. Compared to images processed with NOAA, FXAA, and TAA, the overall image appears most realistic, with superior detail presentation.

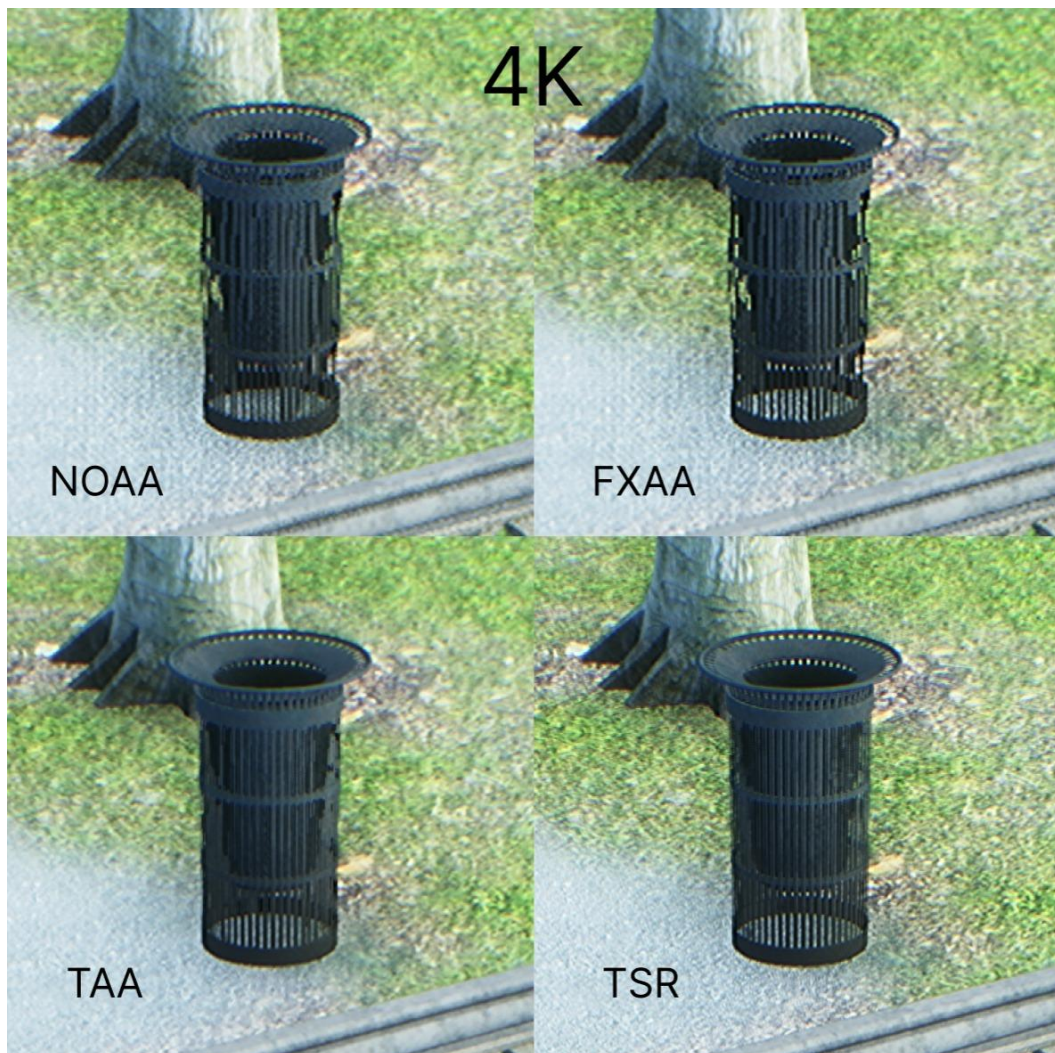


Figure 5.4.2 Trash can images with different anti-aliasing techniques 4K

From the image, we can clearly see that the original image without any anti-aliasing applied (NOAA) still exhibits noticeable “jaggies,” even more so than at

1K resolution. The circular edges of the trash can still show jaggies, and the metal wires in its body appear as a series of stair-step-like line segments. The stone wall background still exhibits stair-step edges, with a grainy texture and lack of clarity;

After applying Fast Approximate Anti-Aliasing (FXAA), the most noticeable improvements are the reduction in stair-step textures and curved edges along diagonal lines. The edges of the trash can and the contours of the stone wall have become significantly smoother. However, the metal wires on the main body of the trash can have become overly blurred, causing adjacent wires to merge into thicker bands, with some areas appearing hollow, resulting in unnatural transitions. Compared to NOAA, the overall improvement is minimal: fewer jagged edges, but reduced clarity in complex meshes.

After applying Time-based Anti-Aliasing (TAA), the image becomes clearer and more continuous. The circular edges of the trash can are almost free of jagged edges, and the metal wires on the can body are largely visible. Due to the effects of temporal reprojection, some minor ghosting or blurring may still remain on the metal wires. The stone wall background displays clear grout lines and distinct textures, with jagged edge artifacts nearly eliminated;

After applying Time Super Resolution Anti-Aliasing (TSR), the image achieves the clearest and most realistic effect. The edges of the trash can and the metal wires on the body are rendered with exceptional precision: no blending occurs. The texture of the stone wall is even more refined, with minimal motion artifacts. Visually, the image processed with TSR balances smooth edges with the retention of fine details.

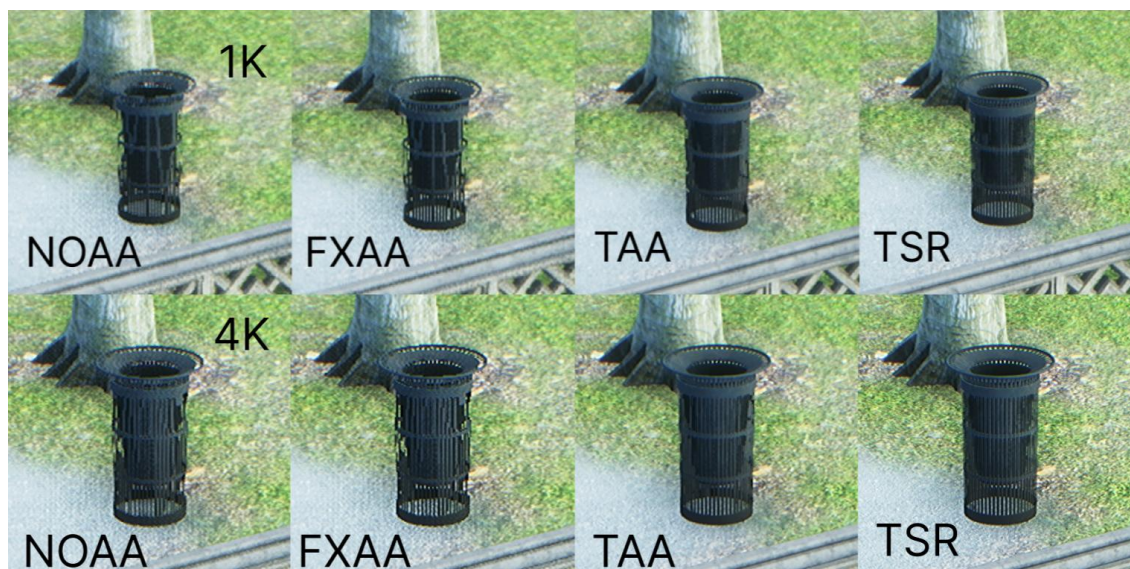


Figure 5.4.3 Comparison of 1K and 4K images of trash can using different AA

As shown in Figure 5.4.3, we compare the four rendering results (NOAA, FXAA, TAA, TSR) at 1K (top row) and 4K (bottom row) resolutions side by side, making it easier to observe the differences in how different anti-aliasing techniques handle the same detail area from a horizontal perspective:

- **NOAA**

Without anti-aliasing, the metal ring of the trash can and the intersecting lines of the bottom grid are noticeably jagged. At 4K, the jaggedness is still visible and even more prominent due to the higher resolution; the stone wall seam edges also have stair-step jaggedness, and the picture is grainy and the details are blurred.

- **FXAA**

FXAA can smooth the roughest jaggedness at both resolutions, but at the expense of some loss of detail and blurred edges of objects. At 1K, the thin wires of the grid begin to merge into thicker strips; at 4K, this blurring is more obvious, and the previously clear gaps and protrusions become soft and distorted.

- **TAA**

TAA can remove jaggedness while retaining more structural details at both resolutions. At 1K, the metal ring is almost jagged but has slight ghosting; at 4K, the ghosting phenomenon is reduced, the grid details are clearer and the edges are smoother, and the texture and lighting transitions of the stone wall are more natural.

- **TSR**

TSR presents the sharpest details at both resolutions. At 1K, the grid lines are clearly separated with almost no ghosting; at 4K, the subtle metal surface texture and highlight details are preserved, the stone wall is clear and natural, and the overall image quality is closest to the real scene.

From the horizontal comparison, it can be seen that FXAA is better at blurring the jagged edges of the image, thereby reducing the visual jagged effect; TAA can strike a balance between smoothness and clarity; and TSR takes into account both edge smoothing and detail retention at high resolution, which can best meet the visual needs at high resolution.

5.4.2 Visual comparison of different anti-aliasing techniques- Part 2

To further verify the universality and consistency of the subjective visual observation results in the first section, this section selects another set of Downtown Alley scenes for supplementary comparative analysis from different perspectives. We examine four representative detail areas (highlighted in yellow) at high magnification to compare the processing effects of each anti-aliasing algorithm on fine structures such as license plate text, air conditioning ventilation grilles, near-view stairs, and far-view stairs. By displaying images of different geometric structures side by side, the purpose is to test whether the performance trends of various anti-aliasing techniques in terms of edge smoothness and detail retention are consistent.

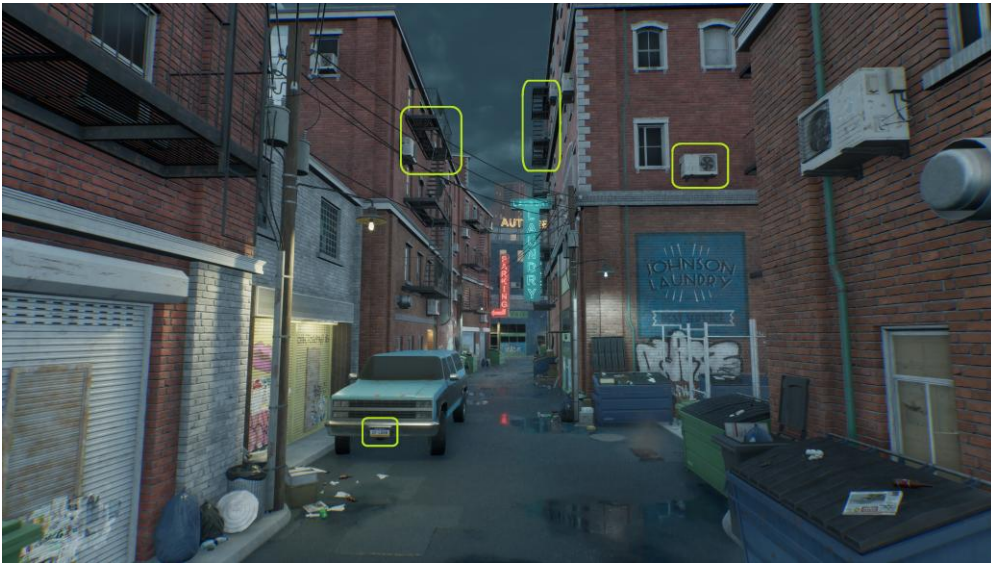


Figure 5.4.4 Comparison of 1K and 4K images of trash can using different AA

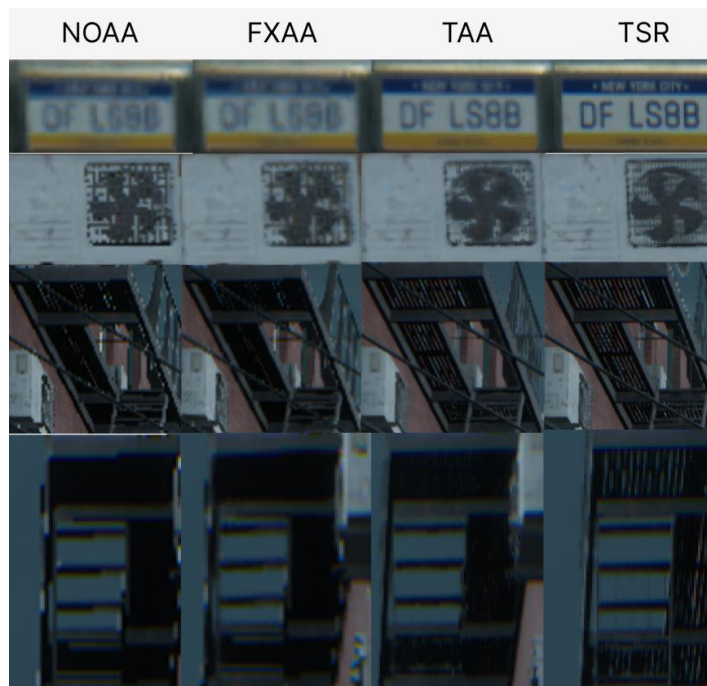


Figure 5.4.5 Comparison of different Anti-Aliasing techniques in detail areas

Detail Area 1: License Plate Characters

- NOAA: The license plate text and the characters in the upper blue area are jagged and pixelated, and the text is difficult to read.
- FXAA: The license plate text and the characters in the upper blue area are smoothed, but the license plate text and the characters in the blue area are still blurry and unrecognizable.
- TAA: The license plate text is partially displayed, but the characters in the blue area are still ghosted and cannot be distinguished.
- TSR: The license plate text is rendered clearly with almost no jaggedness and ghosting, and the letters in the blue area can be distinguished as "NEW YORK DIY".

Detailed area 2: Air conditioning ventilation grille

- NOAA: The thin ventilation grille bars are folded into a jagged shape, and the edges have a serious grainy feeling.
- FXAA: The jaggedness of the bars is reduced, but they are slightly fused, and the sense of separation between the bars disappears.
- TAA: The metal fence bars are visible, but the problem of temporal ghosting in the metal fence bars is more serious, so the lines appear to be fused in the visual effect.
- TSR: The metal fence bars are rendered as continuous and smooth lines, the edges are clearly visible, and the holes formed by the lines are clearly distinguished.

Detailed area 3: Metal Stairs 1

- NOAA: The gaps in the metal filaments appear unevenly distributed due to the presence of jaggedness, and some areas have heavy artifacts.
- FXAA: The artifacts are basically eliminated, but the gaps in the metal filaments are partially fused together to form black areas due to blur processing, and the details cannot be distinguished, with a plastic feel.
- TAA: The gaps in the metal filaments can be basically distinguished, but a small number of detail areas are connected.
- TSR: Improves the detail connection parts in almost all TAA images, and looks the most natural visually.

Detail area 4: Metal staircase 2

- NOAA, FXAA, TAA: The metal staircases have almost the same performance under the three anti-aliasing technologies of NOAA, FXAA and TAA. The metal filaments are completely fused together to form a black area. The metal staircases in the NOAA image have obvious aliasing, while the aliasing phenomenon in the FXAA and TAA images is more blurred.
- TSR: In the case of TSR, the metal filaments of the staircase are clearly visible.

By comparing these four detail comparison images horizontally (Figure 5.3.2), we confirmed the relative performance trend determined in the first part - that is, FXAA achieves anti-aliasing effect by blurring edges, TAA can effectively

reduce aliasing but has motion blur defects, and TSR optimizes many details based on TAA. Later, we will analyze this result in combination with the above image quality metrics and performance metrics in the discussion.

5.5 Discussion

This section summarizes the experimental results of this study's anti-aliasing technology in terms of image quality, performance indicators, statistical analysis, and subjective visual evaluation, and systematically discusses three mainstream anti-aliasing technologies (FXAA, TAA, and TSR), aiming to provide a scientific and detailed empirical basis for subsequent technology selection.

1. Objective comparison of image quality and data changes

At 1K and 4K resolutions, a horizontal comparison of three anti-aliasing technologies, NOAA, FXAA and TAA, showed that TAA performed best in terms of PSNR, SSIM and edge preservation. For example, at 4K resolution, the average PSNR of TAA was 32.52, significantly higher than FXAA (28.26) and NOAA (28.11), an increase of about 15% and 16% respectively. In terms of SSIM, the average TAA was 0.88, an increase of about 11% compared to FXAA (0.79) and NOAA (0.79). In terms of edge preservation, TAA (0.73) was also better than FXAA (0.63) and NOAA (0.63), with an increase of nearly 16%.

The overall performance of TAA shows that it can effectively eliminate aliasing while better preserving image structure and detail information. Although FXAA has advantages in performance, it is slightly inferior to TAA in all image quality indicators, and is only slightly better than NOAA without anti-aliasing. These comparison results further support the rationality of TAA as the mainstream anti-aliasing technology from a data level.

2. Performance Index Analysis and Comparison

In terms of performance, FXAA achieved the highest average frame rate and the lowest GPU rendering time in all test groups. At 4K resolution, FXAA average FPS is 158.55, GPU rendering time is 5.63ms, TAA average FPS is 144.18, GPU time is 6.92ms, while TSR average FPS is only 127.21, GPU time is as high as 8.14ms. Compared with FXAA, TSR's average frame rate is reduced by about 20%, and GPU rendering time is increased by about 44%. TAA is between the two, with a frame rate drop of about 9% compared to FXAA, but an increase of 13% compared to TSR. This shows that TSR brings obvious performance consumption for extreme image quality, while FXAA is suitable for scenes with extreme performance requirements or limited hardware resources.

3. Statistical analysis process and significance test

To further verify whether the differences in the above objective indicators are statistically significant, the paper conducted a one-way ANOVA on the three quality indicators of PSNR, SSIM, and edge preservation. After excluding the interference of the TSR benchmark, the F value of PSNR between TAA, FXAA, and NOAA was 2.72 ($p=0.0805$), which did not reach the statistically significant level ($p>0.05$), but the Kruskal-Wallis non-parametric test results of SSIM and edge preservation ($H=5.561$, $p=0.06202$) were close to the significant level, indicating that different anti-aliasing technologies have certain differences in structural similarity and edge indicators. In terms of performance indicators (FPS, GPU rendering time), the differences between multiple groups did not reach the statistically significant level, indicating that the difference in

performance between FXAA and TAA is mainly reflected in extreme scenes or specific screens.

4. Consistency between subjective visual evaluation and objective indicators

In terms of subjective visual evaluation, TSR has the best image quality and the best detail retention. Whether it is the outline of the object, the details of the material or the overall look and feel, it is closest to reality. TAA has a smooth picture and sufficient anti-aliasing in most scenes, but it is easy to have ghosting and blur in scenes with high-speed motion or drastic changes in light and dark. The sharpness of the FXAA picture decreases most significantly, and the edges are slightly blurred, but the impact on performance is minimal.

5. Experimental process

This experimental process covers the complete process of scene design, image acquisition, automated analysis, statistical testing, data visualization, etc. Through the data collection of multiple sets of scenes with different resolutions and complexities, the image quality and performance of mainstream anti-aliasing technology are fully quantified. However, since the main test scenes are mainly static scenes, the impact of anti-aliasing algorithms on stability under dynamic content cannot be systematically evaluated. Therefore, there are certain experimental deficiencies, and the subjective evaluation samples cannot be included in the large-scale user scoring. In the future, more subjective scores and different hardware platforms can be combined for supplementary verification; and the performance indicators are only collected systematically at 4K resolution, lacking system comparisons at multiple resolutions and different platforms (such as mobile terminals and low-end PCs).

6. Results Summary

In summary, this study systematically quantified the performance of three mainstream real-time anti-aliasing technologies in terms of image quality and performance, revealing the image quality advantage and performance cost of TSR in high-resolution scenes, the good balance between image quality and performance of TAA, and the extreme performance advantage and limited image quality improvement of FXAA. The experimental data and statistical test results provide a scientific and objective reference for the selection of anti-aliasing technology under different project requirements.

6 Conclusion

In this study, we focused on anti-aliasing techniques commonly used in real-time game graphics processing, with Unreal Engine 5 as the experimental platform. By combining fixed camera scene construction and resolution change tests, we conducted experimental comparisons and analyses of three algorithms: FXAA, TAA, and TSR. We tested the performance of different anti-aliasing techniques on PSNR, SSIM, edge retention (image quality metrics), as well as FPS and GPU rendering time (performance metrics), and summarized the image quality performance at 1K/4K resolutions and the performance overhead at 4K resolution. Based on the experimental results, we provide a detailed comparison from the perspectives of image quality and performance consumption, and offer practical recommendations for selecting anti-aliasing techniques.

6.1 Research Summary

Anti-aliasing image quality:

TSR generally performs best in 1080p and 4K image quality metrics, with a clear advantage in maintaining image edge details and overall structural clarity. TAA performs well in smoothing edges and reducing flickering, but suffers from some image blurring and “ghosting” effects. FXAA can blur the jagged edges in images and has the fastest processing speed, but it has significant disadvantages in terms of detail clarity and texture retention.

Performance consumption comparison:

Under the same test conditions, FXAA shows the lowest GPU time and highest frame rate (FPS), offering the best performance and is suitable for mid-to-low-end devices with strict requirements for smoothness. The performance of TAA is between FXAA and TSR. It can achieve better anti-aliasing effect in most cases and is suitable for most mainstream platforms. TSR performs well in terms of image quality, but at the cost of increased GPU utilization and reduced frame rates. Therefore, it should be considered a priority when you have good hardware and require high image quality.

Anti-aliasing selection recommendations:

In extreme image quality and detail performance projects (such as high-tier open worlds, simulations, architectural visualization, or next-gen AAA console and high-tier PC games) we recommend using TSR. Due to its very good detail reconstruction and edge fidelity properties, its use particularly suits adventure, simulation, and cinematic titles that require very high visual fidelity on the proviso there are sufficient hardware resources.

When a project needs image quality and performance balanced (e.g., commercial action, shooting, or massively multiplayer online games requiring smooth frame rates and acceptable visual experience), TAA is optimal. TAA can efficiently suppress flicker and aliasing and minimizes performance loss and can be

employed on e-sports titles, role-playing games, and most commercial PC and console titles.

In extremely hardware-limited situations (e.g., phone games, single-player games on aging GPUs, or games with extremely high real-time requirements for large scope scenes), FXAA is the least problematic solution. It has no significant impacts on speed and is a suitable choice for fast action shooter, platform-hopping, real-time strategy, and other games that require frame rate and compatibility more than image quality.

6.2 Limitations and future research directions

Although this study has built a relatively complete image quality and performance evaluation process, the study still has the following shortcomings:

Limited anti-aliasing coverage: We currently evaluate only Unreal Engine's built-in anti-aliasing methods - FXAA, TAA, and TSR - as a proxy for spatial, temporal, and AI-enhanced methods. However, we do not include the latest super-resolution and hybrid anti-aliasing techniques, such as NVIDIA DLSS, AMD FSR, Intel XeSS, SMAA, MLAA, and hardware acceleration, which are the latest solutions. In the future, we will consider including more modern anti-aliasing and super-resolution algorithms in the evaluation system to expand the depth of research.

Dynamic scenes are not covered: Due to the limitations of the test environment and conditions, the experiments are mainly focused on static scenes, and the impact of anti-aliasing algorithms on image stability under dynamic content (such as motion blur, fast corner points, etc.) is not tested. The effects of fast camera motion, dynamic objects, particle systems, motion blur and other transient phenomena on time stability and ghosting were not tested.

Absence of Subjective User Studies: All experiments in this article are analyzed based on objective metrics and fail to include the user perception experience part. In the future, user subjective scoring can be introduced or a subjective image quality database can be used to further enrich the evaluation system.

Hardware and Platform Diversity: Due to the lack of hardware support, this research was run on a single device configuration equipped with an NVIDIA-class GPU. The performance and quality of the host, mobile GPU, integrated graphics card, and different driver versions were limited by the lack of hardware equipment.

Additional Quality and Performance Metrics: In this study, we only tested five main indicators (PSNR, SSIM, Edge preservation, FPS and GPU Time), and the performance indicators are only collected systematically at 4K resolution, lacking system comparisons at multiple resolutions and different platforms (such as mobile terminals and low-end PCs). Other related quality performance indicators (such as LPIPS, VMAF), memory bandwidth usage, power consumption, latency, and the impact of multithreading on CPU-GPU synchronization have not been tested.

6.3 Future Work Prospects

With the continuous advancement of technology, the future trend is that hybrid solutions are gradually gaining attention. NVIDIA is testing a combination of TAA and FXAA to reduce the HUD flicker problem. At the same time, the introduction of AI-enhanced technologies, such as the "ray reconstruction" function of DLSS3.5, is gradually improving the effect of TAA. It is expected that by 2026, neural network-based temporal anti-aliasing technology will achieve lower latency processing.

In the current market, TAA is gradually becoming the standard for 3A games, with a usage rate of 89%, while FXAA occupies 25% of the lightweight application market with its robustness. Developers should reasonably choose or combine these two anti-aliasing solutions based on the target hardware platform and actual needs to achieve the best visual effect and performance balance.

Expand the scope of anti-aliasing algorithm evaluation: Introduce and compare the latest AI super-resolution and hybrid anti-aliasing technologies such as NVIDIA DLSS, AMD FSR, Intel XeSS, and comprehensively evaluate their performance in image quality and performance.

Temporal stability test in dynamic scenes: Under dynamic conditions such as camera rotation, character movement, particle system and motion blur, measure the temporal artifacts (ghosting) and inter-frame consistency of each technology to supplement the experimental results of static scenes.

Subjective user research and perception index fusion: Conduct user subjective rating (MOS) experiments, collect visual comfort and preference data of different AA schemes, and correlate them with objective indicators (PSNR, SSIM, edge retention) for analysis, and optimize the evaluation index weights.

Cross-platform and multi-hardware verification: Expand to AMD, mobile device integrated graphics cards and host platforms (PlayStation/Xbox), verify the adaptability and performance differences of AA algorithms under different hardware architectures, and form configuration recommendations that are common to multiple platforms.

Enriching evaluation indicators and resource consumption analysis: Adding perceptual quality indicators such as LPIPS and VMAF, as well as measuring performance parameters such as memory bandwidth, power consumption, and rendering latency, to build a multi-dimensional, multi-condition comprehensive evaluation matrix.

By addressing these limitations, future research can build on the repeatable framework proposed here to gain a more comprehensive understanding of real-time AA techniques - balancing objective performance, perceived quality, hardware diversity, and practical usability in dynamic real-world gaming environments.

7 Personal Reflections

When I took the first step of studying abroad, my goal was to complete the master's program. However, the most difficult step for me to complete the master's program is to complete TFM. It is not just as simple as writing a thesis. It is a comprehensive test for me. I remember the freshness when I first entered the campus of the Polytechnic University of Madrid, and I also remember the exhaustion of being filled with courses in the first semester, but now the most important thing is the sense of accomplishment after completing the thesis.

Before choosing the topic of my thesis, I thought deeply about it and finally chose the image processing direction that I am more interested in and have a good course completion. Here, I would like to first thank Professor Alonso Raul and Professor José Crespo for teaching me this course and providing comprehensive guidance on my TFM. Thanks to their patience and support, I was able to apply what I learned in the master's program to my TFM. It is a huge challenge for me to complete this thesis. I not only need to complete the course content that I have hardly touched, but also learn something new. After choosing the direction of anti-aliasing technology, although I was very interested in this direction, I knew very little about the technology in a real sense. Therefore, I spent a lot of time studying the underlying algorithms and application directions of different anti-aliasing technologies. When I first wrote the thesis, it was difficult to proceed because I had little knowledge of the relevant content. After talking with Professor Alonso Raul, he pointed out the direction for me and let me know how to carry out the direction of the entire research, so that I could continue to carry out it wholeheartedly. Of course, the best result now is that I have completed TFM, which I think is the greatest affirmation of myself.

Secondly, I would like to thank my family for their trust and support. It is their support across thousands of miles that has given me the motivation and confidence to come to Spain from China alone to complete the master's project.

I think completing this thesis is not only the end of my master's project, but also the beginning of my next journey. Completing the entire TFM project made me believe that I have the ability to devote myself to a career, to study hard, and to constantly overcome obstacles on the way, so as to continuously improve myself. I gained valuable life experience in this process, which is also a very important lesson in my life. I believe that this experience will provide great help to my future career.

Bibliography

- [1] W. Contributors. "Aliasing". (2025), [Online]. Available: <https://en.wikipedia.org/wiki/Aliasing>
- [2] Sung, K., Pearce, A., & Wang, C. (2002). Spatial-temporal antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 8(2), 144-153.
- [3] Siam, F. J., Prince, Z. I., Bari, A. N., & Uddin, J. (2024). A Real-Time Anti-Aliasing Approach for 3D Applications Using Deep Convolutional Neural Network. <https://cennser.org/IJCVSP/finalPaper/140107.pdf>
- [4] Mengistu, B. (2023). Deep-Learning Realtime Upsampling Techniques in Video Games. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, 10(2), 4.
- [5] Jimenez, J., Gutierrez, D., Yang, J., Reshetov, A., Demoreuille, P., Berghoff, T., ... & Sousa, T. (2011). Filtering approaches for real-time anti-aliasing. *SIGGRAPH Courses*, 2(3), 4.
- [6] Fridvalszky, A., & Tóth, B. (2020). Multisample Anti-aliasing in Deferred Rendering. In *Eurographics (Short Papers)* (pp. 21-24).
- [7] Henry Lin and Andrew Burnes. (2025), "NVIDIA DLSS 4 Introduces Multi Frame Generation & Enhancements For All DLSS Technologies", [Online].Available: <https://www.nvidia.com/en-us/geforce/news/dlss4-multi-frame-generation-ai-innovations/>
- [8] Valve Developer Community. "Anti-Aliasing". (2025), [Online]. Available: <https://developer.valvesoftware.com/wiki/Anti-aliasing>
- [9] Kesten, V. (2017). Evaluating Different Spatial Anti Aliasing Techniques, [Online].Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1106244&dswid=-3533>
- [10] Beets, Kristof, and Dave Barron. "Super-sampling anti-aliasing analyzed." (2000). <http://magicsim.free.fr/3dfx/3dfxpubs/FSAA-Analyzed1.PDF>
- [11] Alex Battaglia. (2024). "Temporal anti-aliasing: a blessing or a curse?". <https://www.eurogamer.net/digitalfoundry-2024-temporal-anti-aliasing-a-blessing-or-a-curse>
- [12] Grahn, A. (2016). An Image and Processing Comparison Study of Antialiasing Methods.
- [13] Zou, J., Xu, Z., Wu, S., & Zhou, L. (2024, March). Improved anti-aliasing algorithm based on fast approximate anti-aliasing. In *Fifteenth International*

- Conference on Graphics and Image Processing (ICGIP 2023)* (Vol. 13089, pp. 568-579). SPIE.
- [14] Reshetov, A. (2009, August). Morphological antialiasing. In *Proceedings of the Conference on High Performance Graphics 2009* (pp. 109-116).
- [15] Jimenez, J., Masia, B., Echevarria, J. I., Navarro, F., & Gutierrez, D. (2018). Practical Morphological Antialiasing. In *GPU Pro 360 Guide to Rendering* (pp. 135-153). AK Peters/CRC Press.
- [16] Laukkanen, S. (2018). Post-processing In video games.
- [17] Zhou, H. (2024, October). The Role of AI in Revolutionizing the Gaming Industry: A Focus on DLSS and Large Language Models. In *2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)* (pp. 939-949). Atlantis Press.
- [18] Epic Games Developers. "Temporal Super Resolution". (2025), [Online]. Available: <https://dev.epicgames.com/documentation/en-us/unreal-engine/temporal-super-resolution-in-unreal-engine>
- [19] Jumani, A. K., Shi, J., Laghari, A. A., Estrela, V. V., Sampedro, G. A., Almadhor, A., ... & ul Nabi, A. (2024). Quality of experience that matters in gaming graphics: How to blend image processing and virtual reality. *Electronics*, 13(15), 2998.
- [20] Goh, E., Al-Tabbaa, O., & Khan, Z. (2023). Unravelling the complexity of the Video Game Industry: An integrative framework and future research directions. *Telematics and informatics reports*, 12, 100100.
- [21] Hore, A., & Ziou, D. (2010, August). Image quality metrics: PSNR vs. SSIM. In *2010 20th international conference on pattern recognition* (pp. 2366-2369). IEEE.
- [22] Sara, U., Akter, M., & Uddin, M. S. (2019). Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, 7(3), 8-18.
- [23] Berg, M. (2024). Energy implications of upscaling and frame generation in games.
- [24] Andreev, D. (2010). Real-time frame rate up-conversion for video games: or how to get from 30 to 60 fps for" free". In *ACM SIGGRAPH 2010 Talks* (pp. 1-1).
- [25] Kato, S., Lakshmanan, K., Rajkumar, R., & Ishikawa, Y. (2011). {TimeGraph}:{GPU} Scheduling for {Real-Time}{Multi-Tasking} Environments. In *2011 USENIX Annual Technical Conference (USENIX ATC 11)*.

- [26] Epic Games Developers. "Anti-Aliasing and Upscaling". (2025), [Online]. Available: <https://dev.epicgames.com/documentation/en-us/unreal-engine/anti-aliasing-and-upscaling-in-unreal-engine>
- [27] Valcasara, N. (2015). *Unreal engine game development blueprints*. Packt Publishing Ltd.
- [28] Epic Games. (2025). *City Park Environment Collection* [Unreal Marketplace Asset]. Unreal Engine Marketplace. <https://www.fab.com/listings/11cc2abb-126c-4452-9fe4-6f2381d96544>
- [29] Epic Games. (2025). *Saloon Interior* [Unreal Marketplace Asset]. Unreal Engine Marketplace. <https://www.fab.com/listings/57991a62-f98f-4c6a-9b94-a74ec12f242e>
- [30] Epic Games. (2025). *Downtown Alley* [Unreal Marketplace Asset]. Unreal Engine Marketplace. <https://www.fab.com/listings/a104cf32-37fe-446f-ac75-cf468d11572a>

Anexos

Folder	Technique	PSNR	SSIM	EdgePreservation	FPS_AVG	GPU_AVG
4K_01	NOAA	25.765	0.788	0.364	213.4	1.8
4K_01	FXAA	25.945	0.795	0.376	204.9	2.0
4K_01	TAA	30.482	0.905	0.597	190.3	4.7
4K_01	TSR	inf	1	1	175.2	4.8
4K_02	NOAA	26.379	0.75	0.415	207.5	1.9
4K_02	FXAA	26.539	0.748	0.408	208.7	2.0
4K_02	TAA	31.298	0.886	0.579	188.7	4.7
4K_02	TSR	inf	1	1	188.0	4.8
4K_03	NOAA	21.879	0.615	0.237	145.3	5.7
4K_03	FXAA	22.251	0.612	0.234	144.9	5.6
4K_03	TAA	26.003	0.794	0.422	119.5	7.2
4K_03	TSR	inf	1	1	98.0	8.9
4K_04	NOAA	26.726	0.75	0.419	149.9	5.4
4K_04	FXAA	26.743	0.745	0.408	152.7	5.3
4K_04	TAA	32.085	0.887	0.607	111.4	7.7
4K_04	TSR	inf	1	1	108.1	7.8
4K_05	NOAA	31.777	0.904	0.757	219.6	3.4
4K_05	FXAA	31.774	0.9	0.739	213.2	3.5
4K_05	TAA	35.557	0.938	0.838	218.2	3.8
4K_05	TSR	inf	1	1	175.4	5.2
4K_06	NOAA	31.835	0.886	0.74	92.7	9.9

4K_06	FXAA	31.85 8	0.88 1	0.727	92.5	9.9
4K_06	TAA	35.06 1	0.92 8	0.8	87.2	10.5
4K_06	TSR	inf	1	1	77.0	12.0
4K_07	NOAA	23.76 1	0.62 4	0.419	178.5	4.5
4K_07	FXAA	23.94 1	0.62 2	0.415	172.2	4.6
4K_07	TAA	28.45 3	0.81 4	0.549	133.6	6.1
4K_07	TSR	inf	1	1	105.6	8.1
4K_08	NOAA	33.75 1	0.89 1	0.811	95.3	9.7
4K_08	FXAA	33.81 2	0.89	0.806	94.8	9.6
4K_08	TAA	32.30 2	0.86	0.787	89.5	10.2
4K_08	TSR	inf	1	1	79.6	11.7
4K_09	NOAA	31.88 8	0.88 3	0.813	98.2	9.2
4K_09	FXAA	33.62 9	0.88 8	0.807	96.8	9.2
4K_09	TAA	36.33 6	0.92 4	0.859	92.1	9.8
4K_09	TSR	inf	1	1	82.5	11.2
4K_01 0	NOAA	32.54 3	0.88 6	0.673	90.5	10.1
4K_01 0	FXAA	31.96 2	0.86 8	0.63	90.7	10.1
4K_01 0	TAA	35.75 1	0.92 6	0.729	79.3	11.6
4K_01 0	TSR	inf	1	1	71.6	13.1
4K_01 1	NOAA	37.28	0.95 2	0.92	235.6	2.4
4K_01 1	FXAA	37.39 5	0.95 2	0.918	233.0	2.5
4K_01 1	TAA	41.2	0.97 2	0.944	224.1	2.9
4K_01 1	TSR	inf	1	1	203.9	4.6

4K_01 2	NOAA	31.28 2	0.9	0.742	204.9	3.2
4K_01 2	FXAA	31.47 7	0.89 6	0.729	198.4	3.3
4K_01 2	TAA	35.11 5	0.94	0.819	196.2	3.7
4K_01 2	TSR	inf	1	1	161.8	5.4