The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-9-2025
30th CIPA Symposium "Heritage Conservation from Bits:
From Digital Documentation to Data-driven Heritage Conservation", 25–29 August 2025, Seoul, Republic of Korea

Comparative Study of Path Tracing Rendering Parameters and Visual Quality in Blender

Etienne Sommer[1], Mathieu Koehl[1], Pierre Grussenmeyer[1]

[1]Université de Strasbourg, INSA Strasbourg, CNRS, ICube Laboratory UMR 7357, 67084 Strasbourg, France
(etienne.sommer, mathieu.koehl, pierre.grussenmeyer)@insa-strasbourg.fr

Keywords: Path Tracing, 3D Rendering, Blender Cycles, Image Processing, Computer Graphics, Medieval Heritage.

Abstract

This study investigates the impact of path tracing samples per pixel count on image quality in Blender, within the framework of 4D reconstructions of partly destroyed castles. These 4D models integrate both spatial and temporal dimensions. High-quality rendering is essential for both research and communication purposes, but it comes with significant computational costs. The study focuses on identifying a balance between visual quality and rendering time by analysing how image quality evolves with the number of rendering iterations.

The evaluation uses the Mean Structural Similarity Index Measure (MSSIM), a perceptual metric that reflects human visual sensitivity more effectively than traditional methods such as the Root Mean Square Error (RMSE) or the Peak Signal to Noise Ratio (PSNR). Each test image is compared to a reference image used as the ground truth. Images are rendered with increasing numbers of samples per pixel while maintaining all other scene parameters fixed to ensure comparability.

The study shows a clear MSSIM convergence, indicating that visual quality improves significantly with more iterations, but converges after a certain threshold. Pixel-wise SSIM maps are also generated to provide local information into the spatial distribution of similarity across the images. In addition, the study examines the role of Blender's built-in denoising algorithms, evaluating their effectiveness in enhancing perceived image quality and their potential to reduce necessary iteration counts.

By quantifying the relationship between samples per pixel and image quality, this research aims to define a rendering strategy for heritage applications. The goal is to minimise rendering time without compromising the visual standards required for documentation, analysis, and public dissemination of digital reconstructions.

1. Introduction

Ray tracing is used to convert a 3D model into a 2D image by casting light rays into a scene. Unlike real-world light, which propagates from light sources to objects, ray tracing follows an inverse approach by casting rays from the camera into the scene. Each ray can then interact with its environment and the materials applied to the 3D models it encounters.

Other rendering methods have been developed to improve the performance and accuracy of ray tracing. One example is path tracing, used by the *Cycles* rendering engine (Blender, 2025) in the Blender modelling software, which allows for the simulation of global illumination effects and the production of photorealistic images. However, it is an iterative method that requires a high number of iterations to reduce noise and achieve a high-quality render. This study aims to analyse the influence of key path tracing parameters, particularly the number of iterations, on the visual quality of the generated image. Here, the practical application is in the context of 4D modelling of partly destroyed medieval heritage, meaning a 3D reconstruction that incorporates a temporal dimension. The objective is to determine the optimal settings that provide a balance between processing time and rendering quality.

This approach is implemented within the framework of the Interreg VI Project (2023-2025): "Châteaux Rhénans - Burgen am Oberrhein", a cross-border initiative involving France, Germany, and Switzerland, coordinated by the European Collectivity of Alsace. Specifically, our methodology aligns with Action 4.6 of the project, which focuses on the 3D valorisation of heritage sites, thereby providing a dynamic and innovative solution for digital heritage conservation and study.

2. Related Work

The use of ray tracing for rendering a 3D scene was introduced by Whitted (1980). The principle involves following the reverse path of light by casting rays from the virtual camera towards the scene to determine interactions with objects. When a ray encounters a surface with an applied material, it reacts according to the material's properties. These can be described by a reflectance model such as the one proposed by Phong (1975), though more advanced models based on the Bidirectional Reflectance Distribution Function (BRDF) also exist. Whitted's model, while relatively simple to implement, has a high computational cost due to the large number of intersections that must be tested, which limits performance. To address this, various methods have been developed to improve the model. One such approach is that of Fujimoto et al. (1986), which accelerates ray tracing by subdividing the scene into a regular grid, thereby reducing the number of intersection tests. Another method, introduced by Kay and Kajiya (1986), employs a hierarchy of bounding volumes (BVH) to further decrease the number of intersection tests and enhance performance.

To overcome some of the limitations of classical ray tracing, improved techniques have been developed, including path tracing, introduced by Kajiya (1986). Its main advantage is the incorporation of global illumination, allowing for a more realistic simulation of light propagation. This method accounts for indirect reflections and maintains energy balance across light bounces, unlike traditional ray tracing, which primarily handles direct and specular reflections. Path tracing involves casting multiple rays in a stochastic manner and accumulating the results. However, when only a small number of samples is used, the final image may appear noisy. Research has been conducted to analyse and compare this noise (Astuti et al., 2022), as well as to reduce it (Parker et al., 2010; Yasenko et al., 2020; Áfra, 2024).

Once the final image is computed, its evaluation generally relies on human visual perception. The challenge then arises: ”How should this image be assessed?” While the question may seem straightforward, it is not easy to answer. Wang et al. (2002) highlight this issue clearly. Wang et al. (2004) propose an initial solution by introducing Structural Similarity Index Measure (SSIM), a method that models human visual perception more effectively than traditional Root Mean Square Error (RMSE) or Peak Signal to Noise Ratio (PSNR) approaches. SSIM makes use of the HSV colour space, which better represents human perception of image quality. Variants of this method have been developed (Wang et al., 2003; Behzadpour and Ghanbari, 2023), along with alternative evaluation techniques based on different principles, such as Visual Information Fidelity (VIF) (Sheikh and Bovik, 2006). Additionally, quality assessment methods can be extended to other types of data, including range images (Malpica and Bovik, 2009) or 3D synthesised views (Battisti et al., 2015).

3. Methodology

A scene was created in Blender to serve as the basis for rendering. It uses the Birkenfels castle, located in Ottrott, around 30 kilometres from Strasbourg, Alsace, France. The castle is thought to have been built in the XIIIth century, then gradually abandoned in the early XVIth century. Figure 1 shows its current state of preservation.

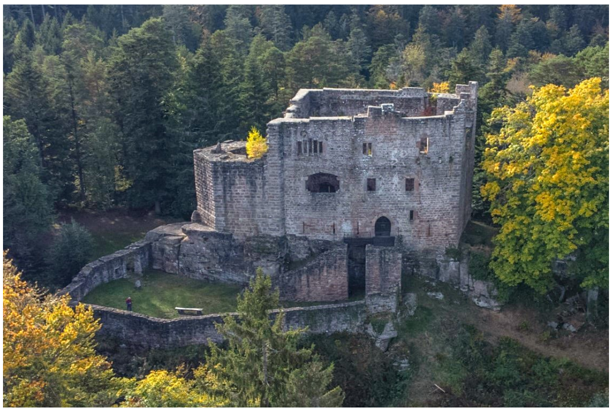

Figure 1. Aerial view of the current state of the Birkenfels castle (Châteaux forts Alsace, 2025)

This scene, presented by figure 2, depicts the historical reconstruction of Birkenfels Castle. It consists of several meshes, with textures generated procedurally. The surrounding terrain is composed of multiple meshes with varying densities, depending on their distance from the castle. Vegetation models are instanced across the terrain: smaller plants are positioned near the castle, while trees populate more distant areas. Lighting is provided by an HDR environment map, and no volumetric effects or fog are present in the scene. All elements were modelled using a parametric approach, as detailed by Sommer et al. (2025). This scene is used to render images using path tracing, with a progressively increasing number of samples per pixel in order to study the visual quality of each rendering. Among the various evaluation methods available, the Mean Structural Similarity Index Measure (MSSIM) was chosen (Wang et al., 2004). This decision is motivated by the fact that MSSIM is widely used in the literature, which facilitates comparison with previous studies and allows the results to be placed in a broader scientific context.

3.1 Generation of the Image Dataset

Each rendered image has a size of 1920×1080 pixels. The evaluation of images rendered using path tracing with varying numbers of samples per pixel is made possible by generating a large number of images, each with a specific samples per pixel count. The rendering of these images is performed directly in Blender with *Cycles* using the Python API. Once an image is rendered with n samples per pixel, a new image is rendered with $n + a$ samples per pixel (a being an integer step value between each samples per pixel count). Depending on the case, each image is then saved either before denoising, or after denoising using *OpenImageDenoise* (OID) (Áfra, 2024), or using *OptiX* (Parker et al., 2010). Each image is saved in the following formats:

- JPG with 100% quality (hereafter referred to as JPG-100)
- JPG with 90% quality (hereafter referred to as JPG-90)
- JPG with 75% quality (hereafter referred to as JPG-75)
- PNG with 8-bit colour depth (hereafter referred to as PNG-8)
- PNG with 16-bit colour depth (hereafter referred to as PNG-16)

For each denoising method, the influence of the number of samples per pixel on the visual quality of the image can then be assessed. This analysis also takes into account the image file format in order to determine whether compression has a significant impact on the final visual quality.

3.2 Creation of a Reference Image

To evaluate the test dataset, a reference image is required to serve as ground truth. This image must have significantly higher quality than the images to which it will be compared. It is therefore generated using the exact same scene and path tracing parameters, but with a number of samples per pixel far greater than that used for the test images. The reference image plays a key role in the evaluation of the results. Its quality has a direct impact on MSSIM scores, which is why it is essential to ensure its high quality. Moreover, it must only serve as a reference for data of the same type, so that the comparisons remain valid. The reference image should approximate a perfect image that has reached convergence, regardless of the number of samples per pixel. This convergence can be assessed using the MSSIM between two images rendered with very high samples per pixel. The closer the MSSIM value is to 1, the better the visual quality. It is therefore necessary to determine a threshold beyond which the rendering converges and reaches a visually satisfactory result. This threshold was estimated at 0.95 by Flynn et al. (2013), based on images assessed by several independent observers. Anglada et al. (2022) adopted this threshold to dynamically adapt scene rendering and accelerate image generation using GPU processing.

Figure 2. Reference image with 100,000 samples per pixel

(a) Without denoising

(b) With OptiX denoising

Figure 3. Comparison between renders with 1 sample per pixel: (a) without denoising and (b) denoised with OptiX.

For the test dataset, achieving an MSSIM > 0.95 typically requires several hundred samples per pixel. Therefore, it can be estimated that a reference image must have at least 10,000 samples per pixel to be considered satisfactorily converged. To validate this value, the MSSIM is calculated between an image rendered with 10,000 samples per pixel and one rendered with 100,000. The resulting MSSIM value exceeds 0.99 regardless of the denoising method, image format, or compression level. Thus, it can be confirmed that from 10,000 samples per pixel onwards, each image may be used as a reference for the test datasets, provided they are of the same type. Unless otherwise specified, each reference image used in the following sections was rendered with 100,000 samples per pixel. Figure 2 shows a reference image rendered with 100,000 samples per pixel and denoised using OpenImageDenoise.

4. Evaluation of Images and Validation of Visual Quality

Each set of images (for example, all images denoised with OpenImageDenoise and saved as JPG with 100% quality) is evaluated against a reference image. An MSSIM value can thus be associated with each image, and therefore with each iteration count of the path tracing render. Figure 3 shows the importance of these iterations. In the absence of denoising, the image appears very noisy and dark because many pixels remain black. The image is completely unusable. In comparison, the denoised image appears much brighter, but very blurred, especially in areas of vegetation. These 2 images have a MSSIM score well below 0.95. For each set of images, when the MSSIM value reaches 0.95, the rendering can be considered visually satisfactory. The faster a denoising method or recording format reaches and exceeds this value, the fewer samples per pixel are required.

4.1 Use of the GPU for Path Tracing Rendering

Blender allows the use of a GPU to accelerate path tracing rendering. For this study, the computer used is equipped with a *Nvidia Quadro RTX 6000* graphics card. This enables the use of the *Cycles Render Device OptiX*. The rendering time is significantly reduced, as shown in figure 4.

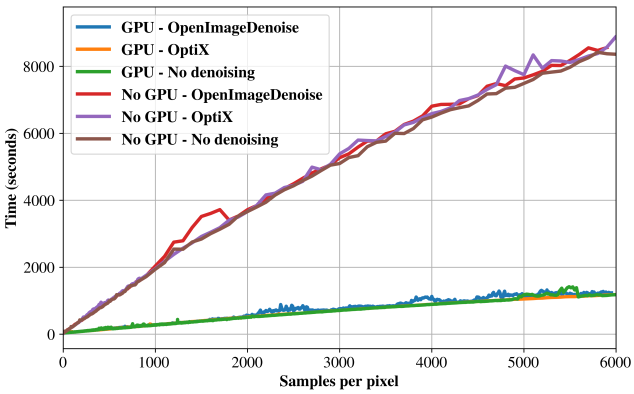

Figure 4. Computing Time using GPU or CPU

However, it is also important to verify whether the use of the GPU affects the final image quality, in order to ensure that the subsequent evaluations are not biased by its use. To assess this, sets of images were rendered both with and without denoising, and both with and without GPU rendering. The images were saved in JPG format with 100% quality, and an MSSIM score was calculated for each image rendered with the same number of samples per pixel. For example, an image rendered with the GPU using 100 samples per pixel and denoised with OptiX was compared to its CPU-rendered counterpart. Figure 5 shows that denoising with OpenImageDenoise gives an MSSIM score above 0.99 after only a few samples per pixel, indicating that rendering with or without the GPU produces visually similar results. The same conclusion holds when no denoising is applied. In contrast, denoising with OptiX produces noticeably different results. For low sample counts (less than 20 samples per pixel), MSSIM scores fall well below the 0.95 threshold required for visual convergence. No specific bibliographic reference was found regarding a more detailed study of OptiX denoising. However, previous research highlights the challenges of denoising highly noisy images resulting from a low number of samples per pixel (Firmino et al., 2022; Sakai et al., 2024). Nonetheless, the 0.95 MSSIM threshold is still reached fairly quickly (after 22 samples per pixel), so this observation regarding low sample counts will not impact the rest of the study.

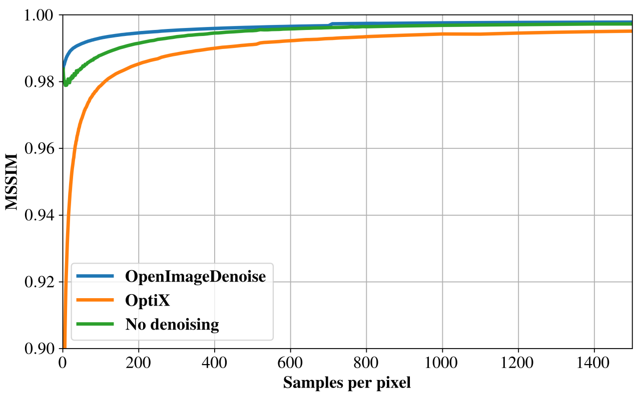

Figure 5. MSSIM comparison between GPU and CPU rendering

In any case, apart from this specific situation, images rendered with the CPU and GPU are very similar. The following analyses will therefore be conducted on images rendered with the GPU, although the results may also be extended to images rendered with the CPU.

4.2 Influence of Image Compression on Visual Quality

When a scene is rendered using path tracing in Blender, the final image can be saved in various formats, with or without compression. Compression allows for significant savings in storage space, particularly when a large number of images is needed, as is often the case when producing a video that requires several hundred or even thousands of frames. For example, for a folder containing 1,315 images, table 1 shows that saving in PNG format with a 16-bits colour depth requires 26 times more storage space than saving in JPG format with 75% quality.

Format	**JPG-75**	**JPG-90**	**JPG-100**	**PNG-8**	**PNG-16**
Size (Go)	0.6	1.0	2.7	7.6	15.2

Table 1. Folder Sizes According to Format and Compression

This compression, while highly beneficial in terms of storage, introduces issues regarding image quality. Saving an image in the .JPG format involves lossy compression, which means that some of the image data is discarded. Specifically, .JPG compression applies a transformation to the image, allowing many coefficients—those close to zero—to be removed after transformation. This transformation, known as the Discrete Cosine Transform (DCT), reduces file size while only slightly degrading the perceived visual quality. High-frequency components of the image can be isolated and removed, as the human eye is less sensitive to them. However, the higher the compression rate, the more the image degrades. As a result, although an image may take up less disk space, it might not be suitable for use if its quality is insufficient.

To analyse the effects of compression, the PNG-16 format is used as a reference. This is a compressed but lossless format, ensuring that the original image quality is preserved. Furthermore, PNG-16 offers higher quality than PNG-8, JPEG-100, JPEG-90, and JPEG-75, as it uses a high colour depth (16 bits per channel), allowing for very fine reproduction of gradients and details, without any data loss. Unlike PNG-8 (limited to 256 colours) and JPEG formats (which are lossy), PNG-16 preserves the full richness of colours, subtle variations, and perfect transparency. By directly analysing images rendered with a very high number of samples per pixel (in this case, 100,000), it is possible to determine whether compressed formats can approximate PNG-16.

Table 2 shows that after 100,000 samples per pixel, saving an image in JPG-75 format does not allow the MSSIM score to exceed 0.95—indicating visual convergence—regardless of the denoising method. Therefore, it can be concluded that beyond a certain level of compression, the image becomes too degraded to maintain sufficient visual quality.

	JPG-75	**JPG-90**	**JPG-100**	**PNG-8**
OID	0,93	0,97	1,00	1,00
OptiX	0,93	0,97	1,00	1,00
No denoising	0,93	0,97	1,00	1,00

Table 2. MSSIM for different denoising methods and image formats with a PNG-16 as reference

Using the same reference image, it is then possible to analyse whether compression during saving affects the speed of convergence and the point at which the MSSIM score exceeds the 0.95 threshold. By iterating over each image with increasing values of samples per pixel, an MSSIM score is calculated for each type of compression, relative to the PNG-16 image rendered with 100,000 samples per pixel. This is done for each format, with the exception of JPG-75, which, as shown in table 2, does not reach the required threshold.

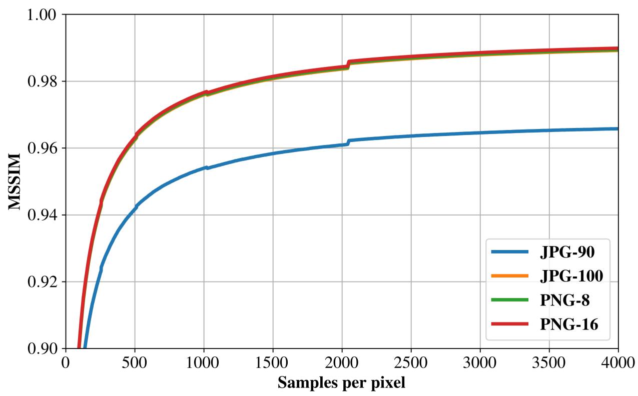

Figure 6. MSSIM score by saving format compared with PNG-16 as reference

It is observed with figure 6 that the JPG-100, PNG-8 and PNG-16 formats converge in a similar manner, while the JPG-90 format requires a greater number of samples per pixel to reach acceptable visual quality. This aligns with the MSSIM scores of images rendered with a similar number of iterations presentend by figure 7: JPG-100 and PNG-8 images achieve similar scores, whereas JPG-90 images have lower scores, indicating that compression slows down and limits convergence towards the ground truth. These results were obtained using OpenImageDenoise for denoising, but similar trends were observed with OptiX and without denoising. It is also interesting that for images saved in JPG-90 format, the MSSIM score shows a decreasing trend. This suggests that at lower values of samples per pixel, the compression ratio has a smaller impact on the MSSIM score.

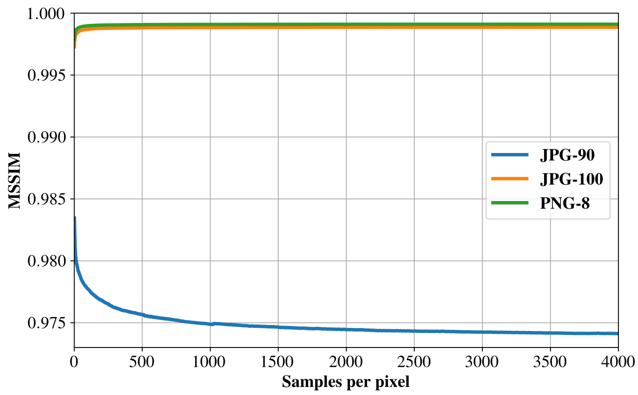

Figure 7. MSSIM score by saving format compared with PNG-16 with the same SPP

It can therefore be concluded that image compression and the number of samples per pixel are interdependent in achieving a result that converges towards the ground truth. If storage space is a critical factor in the project, compression using the JPG format is feasible, provided that a high quality setting is maintained (for instance, 75% quality is insufficient to meet the MSSIM threshold, while 90% quality is adequate). When such compression is used, it becomes necessary to significantly increase the number of samples per pixel, which results in longer rendering times.

Moreover, MSSIM scores between JPG-100, PNG-8, and PNG-16 are very similar. JPG format with 100% quality is a viable option, ensuring optimal visual quality while significantly reducing storage requirements.

4.3 Identifying the Most Effective Denoising Method

The analyses carried out in the previous subsections have helped determine the most suitable methods for image rendering and saving. It was established that GPU-based rendering significantly accelerates processing time while ensuring visual quality comparable to CPU rendering. Furthermore, the JPG-100 format was found to be optimal, as it delivers high visual quality while considerably reducing storage space. Consequently, for the remainder of the analyses, all images are rendered using the GPU and saved in JPG-100 format.

The next objective is to identify the most effective denoising method. The renders were thus performed with OpenImageDenoise, with OptiX and without denoising. As shown in table 2, all three methods converge towards the PNG-16 reference, meaning each method finally achieves a similar final result.

The MSSIM scores, illustrated in figure 8, demonstrate that OpenImageDenoise reaches the MSSIM threshold of 0.95 the fastest, requiring 312 samples per pixel. OptiX closely follows, reaching the threshold at 364 samples per pixel. As expected, in the absence of denoising, significantly lower MSSIM scores are obtained, with at least 532 samples per pixel needed to achieve an MSSIM of 0.95.

The slight fluctuations observed in the MSSIM score curves, such as a drop around 1000 samples per pixel followed by a recovery around 2000 samples per pixel, can be attributed to the irregular behaviour of the path tracing rendering engine. As described by Celarek et al. (2019), certain complex areas of the image, such as soft shadows or reflections, do not always converge in a continuous manner. At 1000 samples per pixel, some lighting effects may appear only partially or in an unstable way, locally altering the image structure compared to the reference, which leads to a decrease in the MSSIM score. At 2000 samples per pixel, these areas begin to converge more accurately, thereby improving the perceived structure and increasing the score. The MSSIM index, being sensitive to local structural organisation, reflects these variations, even if they are not always perceptible to the human eye. In any case, these fluctuations are minor and do not affect the overall interpretation of the results.

5. Repeatability of the Results

The studies and results presented in section 4 are based on a specific scene, observed from a single fixed point of view. As a consequence, the set of elements evaluated is limited to what

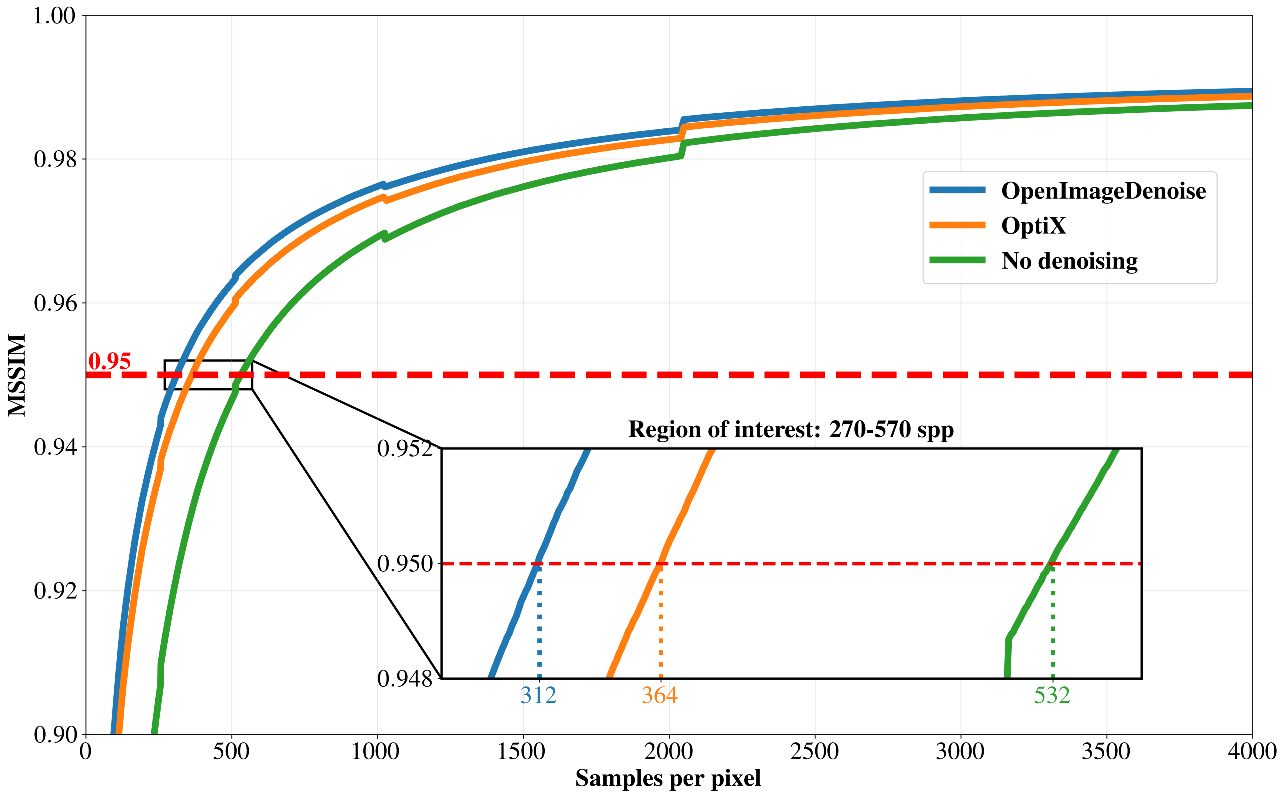

Figure 8. MSSIM score depending on the denoising methods

is directly visible from that particular camera position. While this controlled configuration is necessary to establish initial observations, it also introduces potential biases, as conclusions may only reflect the specific spatial and visual conditions of that viewpoint. It is therefore essential to validate these findings by testing their repeatability across different configurations of the same scene. To evaluate this repeatability, the exact same scene was used, with strictly identical rendering parameters, lighting conditions, 3D models, and material definitions.

Figure 9. Second point of view of the same scene used to study repeatability

Figure 9 shows that the only variation from figure 2 is the position of the camera, offering a different perspective on the same environment. This choice is justified by the scope of the project, which deals with multiple historical castles. These sites often share comparable architectural characteristics and environmental conditions, meaning that findings observed in one setting should be transferable to others. From this new camera position, the rendering pipeline remains strictly consistent with that described previously. The image sequences were generated using the GPU through Blender's Python API, with iterative rendering at increasing sample counts. Each series was produced three times: once with OpenImageDenoise, once with OptiX, and once without any denoising applied.

Before proceeding with the comparison of MSSIM scores, it is important to first consider the potential impact of the new viewpoint on rendering performance. To this end, rendering times were compared between the two viewpoints for equivalent sample counts. For each number of samples per pixel, the rendering time of the second scene was subtracted from that of the first. These time differences are presented in figure 10, and reveal that, apart from a few minor variations, generally under one minute, the rendering times remain very similar across both viewpoints and regardless of the denoising method used.

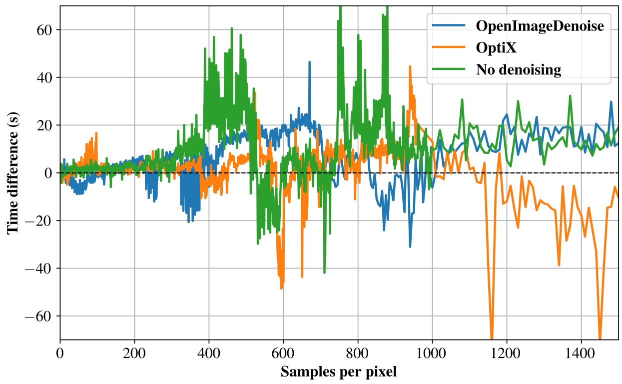

Figure 10. Difference in rendering time between 2 scenes

This consistency confirms that the computational time of the two viewpoints is comparable. For clarity, the figure limits the range of samples per pixel to 1500, but equivalent trends were observed across higher values as well.

The MSSIM results shown in figure 11 were obtained using images saved in .JPG format at 100% quality. A slight variation can be observed in the number of samples per pixel required to exceed the 0.95 MSSIM threshold. These results are presented in table 3.

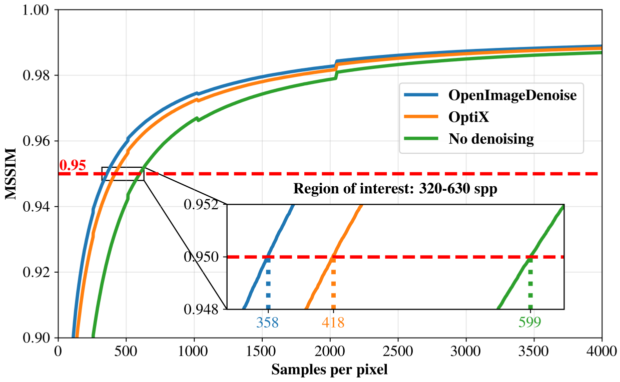

Figure 11. MSSIM score depending on the denoising methods for the second viewpoint

	OID	OptiX	No denoising
Viewpoint 1	312	364	532
Viewpoint 2	358	418	599

Table 3. Samples per pixel required to achieve the MSSIM score of 0.95 for 2 viewpoints

In the first viewpoint, the MSSIM score of 0.95 is reached at 312 samples per pixel with OpenImageDenoise (the results of which are shown in figure 8), while in the second, it is reached at 358 samples per pixel. This difference can be explained by the change in spatial composition resulting from the new camera angle. Certain elements such as areas with indirect lighting or detailed textures can be more prominent or more numerous in the second view, thus requiring a greater number of samples to reduce noise sufficiently. On the other hand, the convergence trend remains very similar from this new point of view: denoising with OpenImageDenoise is much more effective than with OptiX, or even without denoising. It is also interesting to note that similar peaks appear around 1000 and 2000 samples per pixel. In addition to the hypothesis suggested in subheading 4.3, Cycles could include some specific variations when these samples per pixel values are reached regardless of the scene configuration. This is not central to this study and does not change the interpretation of the results, but a more in-depth study of this phenomenon would certainly be useful to enrich the results presented.

However, these results highlight the fact that a single scene cannot characterise all possible configurations, but they do allow us to give general results for this type of scene, with small variations depending on the exact configuration of the scene.

6. Conclusion

The visual quality of a rendering produced by path tracing using the Cycles rendering engine in Blender improves as the number of samples per pixel increases. This improvement has been quantified using the MSSIM score, by comparing a large number of images to a reference image. The results presented show that several factors contribute to the final image quality, which is considered satisfactory when the MSSIM score exceeds 0.95.

The image output format plays an important role: regardless of the number of samples per pixel used for rendering, excessive compression during saving prevents convergence toward the reference image and fails to reach an acceptable visual quality threshold. However, saving in JPG format at 100% quality allows one to achieve MSSIM scores close to those of lossless formats such as PNG, while significantly reducing storage requirements. Moreover, denoising techniques make it possible to reduce the number of samples per pixel needed to achieve satisfactory results, thereby saving rendering time. Denoising can almost divide by 2 the number of samples per pixel required to reach an MSSIM score of 0.95. Comparisons between the two denoising algorithms, OpenImageDenoise and OptiX, have shown that the first one achieves the 0.95 MSSIM threshold with slightly fewer samples per pixel, and performs significantly better when rendering with a very low number of samples per pixel.

Finally, rendering the same scene from a different viewpoint has demonstrated the repeatability of the results. For identical models and materials, we can see that the results are similar between the 2 different points of view. The rendering time is similar for both, and despite some differences in the number of samples per pixel required to reach the MSSIM threshold of 0.95, the best results obtained with OpenImageDenoise have been confirmed on this second point of view. These results give an order of magnitude for path tracing rendering of scenes of this type. It appears that rendering with approximately 350 samples per pixel, denoised with OpenImageDenoise, and saved in JPG-100 format, is optimal for ensuring high visual quality while minimizing rendering time and storage requirements.

It is nevertheless worth noting that these results may not be directly applicable to all situations. A study involving a completely different scene featuring, for instance more reflections, shadowed areas, point lights, or complex geometries, could lead to different conclusions. Furthermore, this analysis could be extended by examining more specific path tracing parameters, such as the influence of path guiding or the number of light bounces, which may further accelerate rendering performance.

Acknowledgement

We extend our thanks to the ”Châteaux Rhénans: Burgen am Oberrhein” Interreg VI Project (2023-2025) involving France, Germany, and Switzerland, coordinated by the European Collectivity of Alsace. We appreciate the opportunity to contribute to this innovative approach to digital heritage conservation.

References

Anglada, M., de Lucas, E., Parcerisa, J.-M., Aragón, J. L., González, A., 2022. Dynamic sampling rate: harnessing frame coherence in graphics applications for energy-efficient GPUs. *The Journal of Supercomputing*, 78(13), 14940–14964. https://doi.org/10.1007/s11227-022-04413-7. Place: USA Publisher: Kluwer Academic Publishers.

Astuti, I. A., Purwanto, I. H., Hidayat, T., Satria, D. A., Haryoko, Purnama, R., 2022. Comparison of Time, Size and Quality of 3D Object Rendering Using Render Engine Eevee and Cycles in Blender. *5th International Conference of Computer and Informatics Engineering (IC2IE)*, IEEE, Jakarta, Indonesia, 54–59.

Battisti, F., Bosc, E., Carli, M., Callet, P. L., Perugia, S., 2015. Objective image quality assessment of 3D synthesized views. *Signal Processing: Image Communication*, 30, 78–88. https://doi.org/10.1016/j.image.2014.10.005.

Behzadpour, M., Ghanbari, M., 2023. Improving precision of objective image/video quality meters. *Multimedia Tools and Applications*, 82(3), 4465–4478. https://doi.org/10.1007/s11042-022-13416-8.

Blender, 2025. Cycles render engine. `https://www.cycles-renderer.org/`. Online: 2025-02-12.

Celarek, A., Jakob, W., Wimmer, M., Lehtinen, J., 2019. Quantifying the Error of Light Transport Algorithms. *Computer Graphics Forum*, 38(4), 111–121. https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13775.

Châteaux forts Alsace, 2025. Châteaux forts Alsace - Château fort. `https://www.chateauxfortsalsace.com/fr/ressources-pedagogiques/chateau-fort/`. Online: 2025-05-30.

Firmino, A., Frisvad, J., Jensen, H., 2022. Progressive Denoising of Monte Carlo Rendered Images. *Computer Graphics Forum*, 41(2), 1–11. https://doi.org/10.1111/cgf.14454.

Flynn, J. R., Ward, S., Abich, J., Poole, D., 2013. Image Quality Assessment Using the SSIM and the Just Noticeable Difference Paradigm. D. Harris (ed.), *Engineering Psychology and Cognitive Ergonomics. Understanding Human Cognition*, Springer Berlin Heidelberg, Berlin, Heidelberg, 23–30.

Fujimoto, A., Tanaka, T., Iwata, K., 1986. ARTS: Accelerated Ray-Tracing System. *IEEE Computer Graphics and Applications*, 6(4), 16–26. https://doi.org/10.1109/MCG.1986.276715.

Kajiya, J. T., 1986. The rendering equation. *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’86, Association for Computing Machinery, New York, NY, USA, 143–150.

Kay, T. L., Kajiya, J. T., 1986. Ray tracing complex scenes. *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’86, Association for Computing Machinery, New York, NY, USA, 269–278.

Malpica, W., Bovik, A., 2009. Range image quality assessment by Structural Similarity. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taïwan, 1149–1152.

Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M., 2010. OptiX: a general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4), 1–13. https://doi.org/10.1145/1778765.1778803. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Phong, B. T., 1975. Illumination for computer generated pictures. *Communications of the ACM*, 18(6), 311–317. https://doi.org/10.1145/360825.360839. Publisher: Association for Computing Machinery.

Sakai, H., Freude, C., Auzinger, T., Hahn, D., Wimmer, M., 2024. A Statistical Approach to Monte Carlo Denoising. *SIGGRAPH Asia 2024 Conference Papers*, SA ’24, Association for Computing Machinery, New York, NY, USA. event-place: Tokyo, Japan.

Sheikh, H., Bovik, A., 2006. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2), 430–444. https://doi.org/10.1109/TIP.2005.859378.

Sommer, E., Koehl, M., Grussenmeyer, P., 2025. Parametric Modelling Techniques for Rhine Castle Models in Blender. *Heritage*, 8(1). https://www.mdpi.com/2571-9408/8/1/31.

Wang, Z., Bovik, A. C., Lu, L., 2002. Why is image quality assessment so difficult? *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4, Orlando, FL, USA, IV–3313–IV–3316.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. https://doi.org/10.1109/TIP.2003.819861.

Wang, Z., Simoncelli, E., Bovik, A., 2003. Multiscale structural similarity for image quality assessment. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2, IEEE, Pacific Grove, CA, USA, 1398–1402.

Whitted, T., 1980. An improved illumination model for shaded display. *Communications of the ACM*, 23(6), 343–349. https://doi.org/10.1145/358876.358882. Publisher: Association for Computing Machinery.

Yasenko, L., Klyatchenko, Y., Tarasenko-Klyatchenko, O., 2020. Image noise reduction by denoising autoencoder. *IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kiev, Ukraine, 351–355.

Áfra, A. T., 2024. Intel\textsuperscript® Open Image Denoise. `https://www.openimagedenoise.org/`. Online: 2025-06-19.