applied sciences

MDPI

Article

Deep Learning-Based Denoising for Interactive Realistic Rendering of Biomedical Volumes

Elena Denisova [1], Leonardo Bocchi [2] and Cosimo Nardi [3,*]

[1] Imaginalis s.r.l., 50019 Sesto Fiorentino, Italy; e.denisova@imaginalis.it
[2] Department of Information Engineering, University of Florence, 50139 Florence, Italy; leonardo.bocchi@unifi.it
[3] Radiodiagnostic Unit n. 2, Department of Experimental and Clinical Biomedical Sciences, University of Florence, Careggi University Hospital, 50134 Florence, Italy
* Correspondence: cosimo.nardi@unifi.it

Abstract

Monte Carlo Path Tracing (MCPT) provides highly realistic visualization of biomedical volumes, but its computational cost limits real-time interaction. The Advanced Realistic Rendering Technique (AR^{2}T) adapts MCPT to enable interactive exploration through coarse images generated at low sample counts. This study explores the application of deep learning models for denoising in the early iterations of the AR^{2}T to enable higher-quality interaction with biomedical data. We evaluate five deep learning architectures, both pre-trained and trained from scratch, in terms of denoising performance. A comprehensive evaluation framework, combining metrics such as PSNR and SSIM for image fidelity and tPSNR and LDR-ꟻLIP for temporal and perceptual consistency, highlights that models trained from scratch on domain-specific data outperform pre-trained models. Our findings challenge the conventional reliance on large, diverse datasets and emphasize the importance of domain-specific training for biomedical imaging. Furthermore, subjective clinical assessments through expert evaluations underscore the significance of aligning objective metrics with clinical relevance, highlighting the potential of the proposed approach for improving interactive visualization for analysis of bones, joints, and vessels in clinical and research environments.

Keywords: Monte Carlo Path Tracing; deep learning noise reduction; biomedical volumes

Academic Editor: Thomas Lindner

Received: 25 July 2025
Revised: 5 September 2025
Accepted: 8 September 2025
Published: 9 September 2025

Citation: Denisova, E.; Bocchi, L.; Nardi, C. Deep Learning-Based Denoising for Interactive Realistic Rendering of Biomedical Volumes. *Appl. Sci.* **2025**, *15*, 9893. https://doi.org/10.3390/app15189893

1. Introduction

Given that hardware performance has reached a certain threshold, three-dimensional (3D) rendering of biomedical volumes is gaining popularity, particularly among the younger generation of clinicians who actively integrate cutting-edge technologies into their daily routines. The use of 3D representation has demonstrated its utility in quickly comprehending traumas in anatomically complex areas, especially by inexperienced physicians and patients, aiding in surgical planning, simulation, and training, as shown by Bueno et al. [1] and Ebert et al. [2].

Dappa et al. [3] demonstrated that among the methods offering 3D visualization of biomedical volumes, Monte Carlo Path Tracing (MCPT)-based algorithms produce images with higher naturalism and clinical value. Recently, Denisova et al. [4] introduced the Advanced Realistic Rendering Technique (AR^{2}T), inspired by MCPT and applied to biomedical volumes. The method allows interaction with data of low quality, such as producing images with a high level of noise, inherent to these types of algorithms. Interactivity is achieved with just one iteration of the algorithm, generating a single sample-per-pixel (SPP) for preview. Once the interaction is complete, and the mouse button (or

wheel) is released, ten iterations of the AR^{2}T are executed, producing a better 10 SPP image. To enhance quality during interactions, a Gaussian blur filter is applied as a post-processing step in each iteration, making the overall image more understandable. However, the images still suffer from noise and a low level of detail, requiring at least a couple of hundred iterations and several seconds for an acceptable result, as shown in Figure 1.

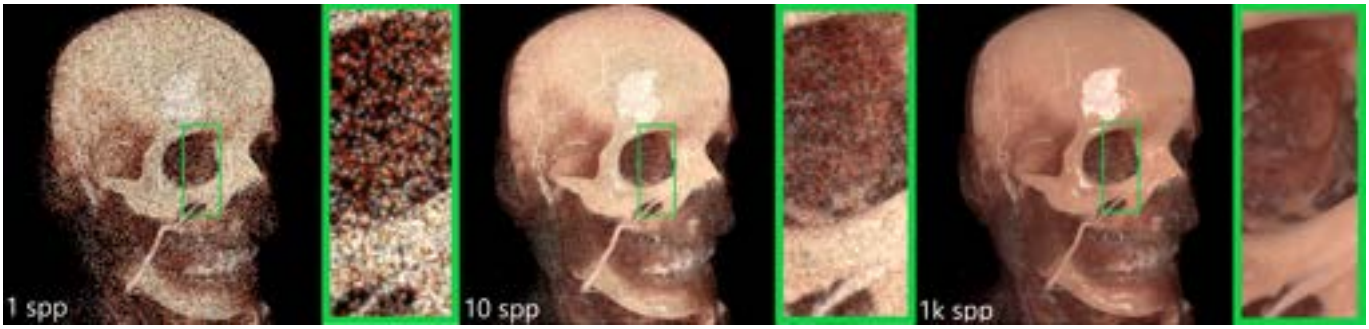

Figure 1. An example of the images progressively generated by AR^{2}T [4] for Manix spiral CT scan. Gaussian blur filter is applied on a single sample per pixel (1 SPP) and 10 SPP images; the reference image is obtained on the algorithm convergence (1000 SPP).

The idea of applying noise reduction algorithms to MCPT is as old as the solution to the rendering equation itself proposed by Kajiya in 1986 [5]. The stochastic nature of the Monte Carlo simulation makes noise inevitable in early iterations and in areas with a lack of lighting, making noise a principal drawback of the method. The more complex the scene and its lighting configuration, the greater the number of iterations needed to produce a clear, noise-free image. Noise reduction, as a post-processing pass applied to the not fully converged image, can significantly reduce the time for generating an acceptable, albeit not always correct, result.

Originally, non-linear filters were applied locally to stochastically sampled images to eliminate spike noise. Recently, machine learning-based techniques have proven more effective in the context of noise reduction for MCPT—both for predicting filter parameters and generating denoised images directly. The results of deep learning approaches are impressive, but usually, in the gaming or movie industry, machine learning is applied for surface-only rendering. Denoising filters can sometimes misinterpret fine details in the medium as noise, leading to their unintentional removal. Alternatively, they may preserve surface edges, making them excessively sharp when observed through volumes, as noted by Hofmann et al. [6].

In this study, we investigate the application of deep learning models—both pre-trained and trained from scratch—for the AR^{2}T [4] denoising during interaction with data. Each architecture was pre-trained independently for two scenarios: (1) using 1 SPP inputs to enable data interaction and (2) using 10 SPP inputs for higher-quality rendering upon interaction cessation. These models were evaluated under their respective conditions to provide a direct comparison of performance tailored to each sample density.

To address the memory constraints of processing large biomedical volumes, we introduce a GPU-efficient, tile-based denoising approach. This technique ensures scalable performance, making it feasible to handle multi-gigabyte medical datasets and facilitating broader applicability to post-processing tasks in biomedical visualization.

We evaluate the models using quantitative metrics, including the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and LDR-ƎLIP, a perceptual quality metric proposed by Andersson et al. [7]. Temporal stability during interactions is assessed with the *temporal PSNR* (tPSNR) metric, following Hasselgren et al. [8]. Additionally, the clinical

survey, involving over 50 medical experts, provides subjective evaluations of denoising performance, particularly regarding the preservation or removal of critical details.

Our findings reveal that models trained from scratch consistently outperform pre-trained models across all metrics, including PSNR, tPSNR, SSIM, and LDR-ꟻLIP. These findings challenge the conventional reliance on large, diverse datasets for model pre-training, highlighting the importance of training specifically for biomedical imaging applications. Notably, the denoising of MCPT-rendered biomedical images presents unique challenges due to the volumetric nature of the data, where the produced noise differs significantly from surface-based noise in traditional imaging.

Clinical feedback further confirms the high confidence physicians have in the quality of images produced by deep learning-based denoising, despite underscoring the necessity of incorporating objective metrics that better align with qualitative assessments and ensuring that denoising methods meet clinical expectations.

2. Related Work

2.1. Non-Linear Denoising Filters

Despite the development of robust denoising algorithms in image processing, most assume spatially invariant noise, which challenges their application to Monte Carlo rendering.

Efforts to apply non-linear denoising filters to stochastically sampled images began in the early nineties with Lee et al. [9] and Rushmeier et al. [10]. Peng et al. [11] later proposed a denoising technique for surfaces using locally adaptive Wiener filtering. In 2005, Xu et al. [12] extended bilateral filtering to incorporate local adaptive noise reduction. Dammertz et al. [13] introduced an edge-avoiding filtering method using the À-Trous wavelet transform with auxiliary depth and normal buffers. Kalantari et al. [14] explored multilevel denoising in a spatially varying manner, and Rousselle et al. [15] demonstrated the effectiveness of combining color and feature buffers in non-local means and cross-bilateral filtering. Bu et al. [16] addressed denoising of glossy surfaces using impulsiveness maps to filter outliers. Bitterli et al. [17] used auxiliary buffers like normal and albedo for non-linearly weighted first-order regression. More recently, Boughida et al. [18] tailored a non-local Bayesian filter for noise from Monte Carlo rendering.

2.2. Deep Learning-Based Denoising

In 2015, Kalantari et al. [19] identified a correlation between noisy scene data and optimal parameters for non-linear filters, suggesting learning this relationship via a multilayer perceptron neural network. Later, Vogels et al. [20] proposed predicting local reconstruction kernels using kernel-predicting networks, while Bako et al. [21] employed a deep convolutional neural network (CNN) architecture for kernel prediction.

Another direction in deep learning-based MCPT denoising aims to predict per-pixel color directly. Xu et al. [22] proposed an adversarial denoising approach based on a generative adversarial network (GAN), and Wong et al. [23] used a deep residual network (ResNet), showing significant improvements over basic CNNs. Kettunen et al. [24] also applied U-Net autoencoders to gradient-domain rendering.

Most techniques, whether handcrafted or deep learning-based, focus on surface rendering and exclude participating media like clouds, fog, liquids, and medical data, where light transport and scattering among particles complicate denoising. Traditional image-space MCPT denoisers can struggle in these scenarios, as noted by Huo et al. [25].

Recent advancements in MCPT denoising for scenes with both surfaces and semi-transparent volumetric media include Hofmann et al.'s work on a network tailored for cinematic rendering in undersampled images [26]. Although effective, it remains unsuitable

for real-time processing. More recently, Hofmann et al. [6] explored decomposing rendered images into volume and surface layers for spatio-temporal neural denoising of both.

2.3. Undersampled Image Denoising

Kernel-based post-processing methods have successfully denoised highly undersampled images, but their complex network structures hinder real-time interaction. Chen et al. [27] introduced a lightweight cascaded network for denoising single SPP Monte Carlo images, combining pixel and kernel prediction methods. However, their approach struggles with volumetric media denoising and remains relatively slow.

2.4. Temporal Stability

Unlike static image denoising, which processes each frame independently, several methods use information from previous frames to achieve temporal stability. Schied et al. [28] introduced a spatiotemporal variance-guided filtering technique that increases the effective sample count with temporal accumulation and variance estimates. Hasselgren et al. [8] improved temporal stability by extending their U-Net denoiser with temporal feedback. However, these methods degrade in performance in dynamic scenes. Recently, Lee et al. [29] proposed a fast framework combining 1 SPP and temporally accumulated 1 SPP images to estimate kernel maps. While efficient, their approach relies on controlled setups, limiting its effectiveness in dynamic scenarios, such as in MCPT for biomedical imaging.

2.5. Biomedical Image Denoising

In the field of direct volume rendering, Iglesias-Guitian et al. [30] proposed an analytical method based on a denoiser previously used for surface rendering, demonstrating effective denoising of volumetric path tracing images. However, their method neither specifically addressed biomedical imaging nor underwent clinical evaluation in the context of biomedical rendering. The work of Hofmann et al. [26] is the most similar to ours in terms of application and the use of neural networks for denoising. However, it also lacks clinical evaluation and is limited to offline image and video generation.

To the best of our knowledge, no studies have compared the performance of models using the same architecture, both pre-trained and trained, on domain-specific medical MCPT-based imaging. This work aims to address this gap by investigating the benefits of domain-specific training in medical imaging.

3. Materials and Methods

3.1. MCPT Framework

A survey reported by Denisova et al. in [4] showed that AR^{2}T produces the most realistic, diagnostic, and overall best images of biomedical volumes acquired through computed tomography (CT) or magnetic resonance imaging (MRI) compared to images produced by deterministic algorithms.

In this work, we employ their practical framework for visualizing biomedical volumes using AR^{2}T [4]. This framework enables interaction with data of low quality, allowing adjustments of the transfer function, application of clip planes, and configuration of light size and position. It progressively generates a high-quality image upon user request, with the process being stoppable either manually or automatically when the algorithm converges. The convergence criterion is based on the mean squared displacement (MSD) between iterations; when the MSD becomes less than 10^{-7}, the iterations cease, and the method is considered converged. During interactions, a one-sample-per-pixel (1 SPP) image is displayed to the user, and on mouse release, a ten-samples-per-pixel (10 SPP)

image is generated. A Gaussian blur filter is applied for a more uniform representation. Additionally, the framework supports real-time repositioning of the light source, transfer function modifications, and clipping planes. The medium interaction speed ranges from 10 to 20 frames per second (FPS) on Intel Core i5-7600K CPU @ 3.80 GHz, 16 GB RAM, 4 CPU threads, and NVIDIA GTX 1060 with 6 GB VRAM, running Windows 10 Pro (this PC was used for all tests and training reported in the manuscript, unless otherwise specified) depending on volume complexity. In this environment, we are going to replace the Gaussian blur with a deep learning denoiser and assess whether the resulting quality and confidence are acceptable for use in medical practice.

3.2. Network Architectures

When selecting a deep learning architecture for noise reduction, we prioritized two key principles: result quality and denoising speed. Since our aim is to apply denoising during interactions, it is crucial to achieve fast denoising without significantly compromising speed.

We initially considered two simple sequential autoencoder-based architectures inspired by Zhang et al. [31]—*AE Lite* and *AE Full* (see Figure 2)—for their simplicity and denoising speed. Additionally, we explored U-Net-based architectures for their high-quality results: the *Samsung* and *Tyan* models from the NTIRE 2020 Challenge [32], and a U-Net architecture implemented in Intel's Open Denoiser (*OIDN*), trained from scratch. For models trained from scratch, we avoided auxiliary features like albedo and normal, as traditional G-buffer features are less efficient for volumes, as noted by Hofmann et al. [6]. We also excluded the OIDN-based model by Firmino et al. [33] due to its reliance on variance estimation, which made it slower and less effective for low-sample-per-pixel data.

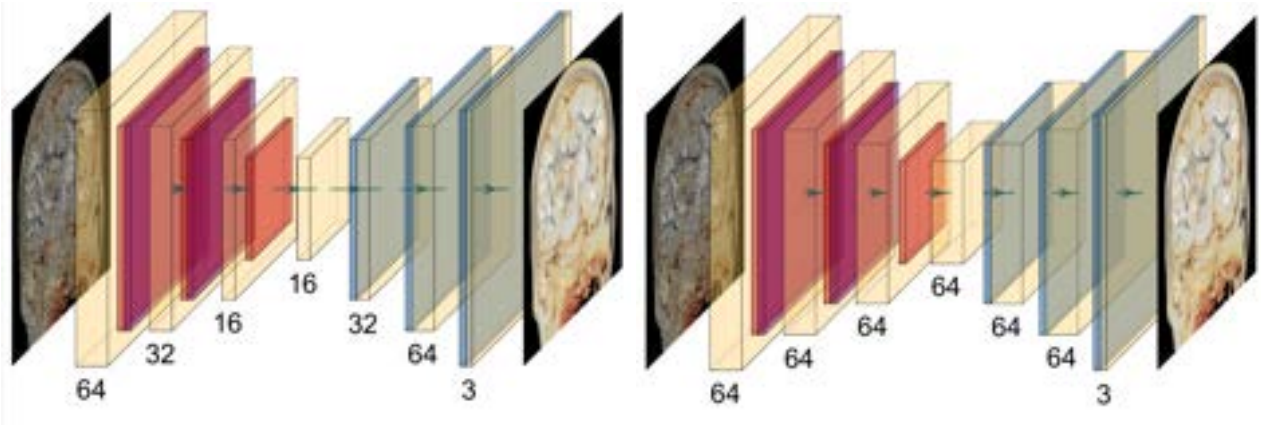

Figure 2. Autoencoder networks: AE Lite (**left**) and AE Full (**right**). The network's input, i.e., noisy image, is on the left, while the network's output is on the right. Convolution (3 × 3) layers are visualized in yellow, pooling (2 × 2) layers in orange, batch-normalization in purple, upscaling (2×) layers in blue. The numbers indicate the number of feature channels at each stage.

To assess the importance of domain-specific training, we incorporated the pre-trained OIDN v.2.1.0 [34] (*OIDN STD**) and its variant with albedo and normal (*OIDN ADV**) (see Figure 3). Here and throughout the text, '*' denotes the pre-trained models.

The choice of the Samsung and Tyan architectures was not justified by its execution time; however, we opted to use them as a reference due to their superior results, as reported in the Challenge [32]. The characteristics of the selected architectures, along with their advantages and disadvantages, are summarized in Table A1.

Figure 3. An example of noisy inputs (1 SPP and 10 SPP), optional albedo and normal auxiliary features, and reference output.

It is worth noting that at the time of our study, our primary aim was to explore architectures that balanced quality and denoising speed, focusing on models that could be implemented and trained efficiently with the available computational resources. We selected architectures such as autoencoders and U-Nets, which provided a practical starting point for evaluating MCPT-specific noise reduction. Additionally, the OIDN architecture served as a well-established benchmark for path-traced data.

Moreover, we prioritized static-frame denoising methods over temporal approaches to meet interaction requirements and ensure consistency between interactive previews and stop-rendered frames, thereby avoiding temporal artifacts. Temporal methods, while beneficial for frame-to-frame consistency, are less suitable for interactive scenarios like ours, where clip planes can dynamically slice through volumes, and rapid changes in light position and transfer function often occur. Our primary aim was to ensure that neural network-based denoising delivers acceptable image quality for physicians, even at very low sample counts, rather than prioritizing smooth frame transitions.

To ensure a fair comparison of denoising performance and execution speed across different architectures, we opted to train all models from scratch using TensorFlow v.2.14.0 [35]. This decision was motivated by the need to standardize the training process, eliminating variations introduced by differences in pre-trained weights, optimization techniques, or framework-specific implementations. By training the models in a unified environment and under consistent conditions, we could accurately assess their relative strengths and weaknesses in terms of both denoising quality and computational efficiency. After completing the training, we converted the saved model to the Open Neural Network Exchange (ONNX) format and employed ONNX Runtime v.1.15.0 [36] for denoising execution.

For OIDN STD* and OIDN ADV*, we employed the OIDN C++ library. Unfortunately, we could only run it on the CPU, as our video card architecture is not supported by an actual OIDN version for GPU execution.

3.3. Image Generation

We generated training images primarily using Cone-Beam Computed Tomography (CBCT) volumes from different animals' and humans' body districts, acquired by SeeFactorCT3^{TM} (human) and Vimago^{TM}GT30 (vet) Multimodal Medical Imaging Platforms. The datasets are publicly available on Kaggle [37]. Additionally, one spiral computed tomography (CT) volume [38] and one magnetic resonance imaging (MRI) volume [39] were included.

To represent more complex internal structures, some volumes were cropped along the sagittal axis, incorporating different transfer functions. The same volume underwent a 10° rotation around the vector $(1, 1, 1)$ and was scaled by 0.02, totaling 36 times. This data augmentation procedure allowed capturing the same anatomical districts with varying levels of detail from different viewpoints and subsequently illuminated in different ways. After each transformation, images were generated until convergence (600–2000 SPP), with the light position fixed and placed in front of the volume.

Monte Carlo-rendered image data typically exhibit high dynamic range (HDR), which can cause instability during training, as noted by Wong et al. [23]. Baco et al. [21] demonstrated that logarithmic normalization can significantly reduce the range of color values of HDR images and improve denoising results. However, given that our image data range between 0 and 3 (with higher values truncated by the generating framework during interactions to mitigate fireflies), we opted to save training images in low dynamic range (LDR) as 8-bit integer values. This was achieved by applying gamma correction first, then truncating values superior to 1, and finally, multiplying by 255, following an approach outlined by Jain et al. [40]. This decision reduced hardware demands and accelerated training. Consequently, three LDR images were saved per volume configuration: 1 SPP, 10 SPP, and the converged reference.

3.4. Dataset Preparation

Every generated image was divided into pieces of size 256×256. Images containing more than 50% black pixels (evaluated using their reference counterparts) were excluded. The remaining images were randomly distributed between the training dataset (85%) and the testing dataset (15%).

3.5. Training Setup

We established two separate training setups: one with the input image of 10 SPP and another with the input image of 1 SPP. We initiated training with 10 SPP, running it for a maximum of 500 epochs with early stopping enabled and a patience of 5 epochs. As a loss function, we utilized the mean squared error (MSE) following Zhang et al. [31], and employed the Adam optimizer [41]. The learning rate was set to 0.001, determined to be optimal by running the training on a reduced dataset for 50 epochs, with the learning rate scheduled to increase from 1×10^{-6} to 0.1. The batch size ranged from 1 to 32, depending on the architecture complexity.

3.6. Denoising on GPU

As AR^{2}T [4] runs on GPU, it can happen (as in our case) that the biomedical volume, several gigabytes in size, together with the denoising model, occupies a major part of the GPU memory, and denoising of fullscreen images of size 1920×1080 has to run on CPU. To address this, we decided to denoise the image in pieces. We split the entire image into smaller images of size 256×256, ensuring that every next piece has an 8-pixel border with adjacent pieces. When reconstructing the fullscreen image, we discard the 8-pixel border. This approach eliminates denoising artifacts that appear on image borders when passing through convolution layers (see Figure 4). This method, while tailored for this specific case, is theoretically applicable to any 2D or 3D image processing task involving neural networks, where memory constraints are critical.

3.7. Quantitative Evaluation

Consistent with the works of Chen et al. [27] and Hoffman et al. [26], we assessed the denoising results quantitatively using the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM), introduced by Wang et al. [42]. PSNR measures how well the denoised image approximates the reference image, with a higher PSNR indicating better quality. However, PSNR alone may not guarantee perceptual quality. On the other hand, SSIM is based on the structural information of the scene and, according to Wang et al. [43], provides a better approximation of perceived image quality.

Moreover, as we are dealing with a medical application, we also incorporated the perceptual LDR-ꟻLIP metric proposed by Andersson et al. [7]. ꟻLIP is a difference evaluator

widely used in image processing evaluations, particularly focusing on differences between rendered images and their corresponding ground truths.

To evaluate the temporal stability of denoising during interactions, we utilized the *temporal PRSN* (tPSNR) metric, which computes the PSNR on the temporal finite differences between consecutive frames, as proposed by Hasselgren et al. [8]. To ensure a fair comparison between different denoising algorithms, we acquire a sequence of noisy frames during interaction and denoise them offline to guarantee that the metrics are computed on the same sequence.

In this study, SSIM was calculated using the **structural_similarity** function from the **scikit-image** Python library (ver. 0.22.0); meanwhile LDR-ꟻLIP was calculated using the **evaluate** function from the **flip** Python library (ver. 1.4).

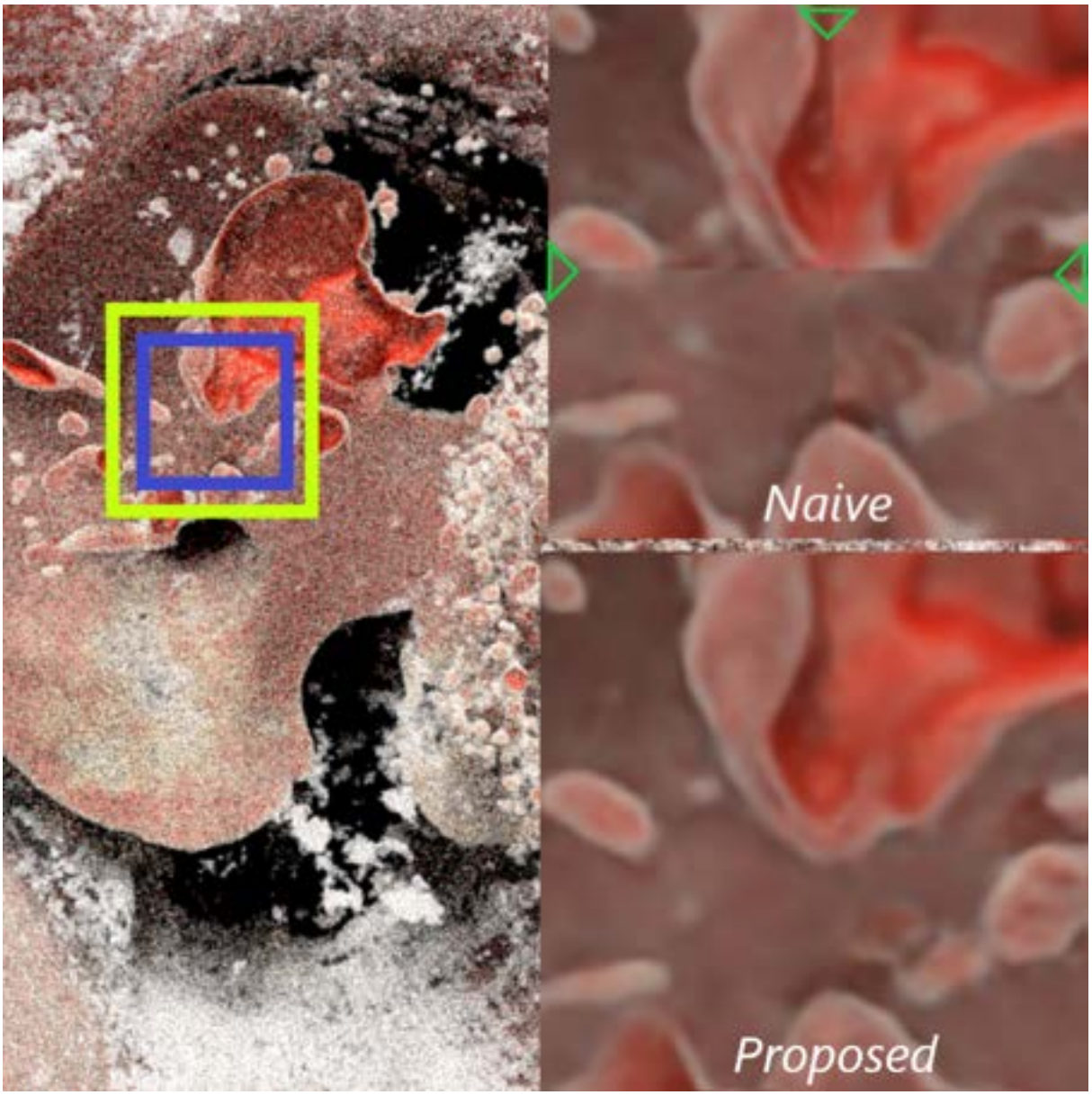

Figure 4. A demonstration of denoising on GPU under limited memory conditions. Larger image is split into 256 × 256 pieces for denoising. External light-green square represents the noisy piece of size 256 × 256 to denoise, while internal blue square is the cropped denoised piece of size 240 × 240 (excluding 8-pixel borders to avoid artifacts at the borders). On the **top-right**, an example of the naive piece-by-piece denoising and reconstruction; on the **bottom-right** is our approach. Green arrows indicate the visible stitching lines observed with the naive approach.

3.8. Visual Assessment

Quantitative evaluations, even based on ꟻLIP or similar metrics, often fall short in capturing the entirety of human perception, especially when dealing with biomedical images. Furthermore, we sought to evaluate denoising performance, focusing specifically on the preservation or removal of critical details. To achieve this, a survey was conducted, where medical experts assessed various aspects of the denoised images.

Initially, experts were asked to choose between undersampled images (1 SPP and 10 SPP) and images generated with OIDN. Then, participants were asked to rate their confidence on a scale from 1 (*Not at all confident*) to 5 (*Extremely confident*) when comparing images generated from 1 SPP and 10 SPP using OIDN and OIDN ADV*. Their choice should be based on the comparison with the converged reference. Subsequently, participants assessed videos of interactions with volumes without AI, with OIDN, and with OIDN ADV*, rating from 1 (*I can't distinguish anything, this preview does not give me any information*) to 3 (*I can distinguish a lot of details, I would be satisfied having this preview*). Participants were also asked to provide comments justifying their choices.

4. Experiments

4.1. Training

A total of 4567 × 3 images (1 SPP, 10 SPP, and reference) of size 1920 × 1080 were generated. After splitting into pieces of size 256 × 256 and excluding images containing more than 50% black pixels, 85% of the remaining 63751 × 3 images were used for training (see Figure 5). A batch size of 32 was set for AE Full and AE Lite, 2 for Tyan and OIDN, and 1 for Samsung. The choice of batch size depended on the model's architecture size, as some did not fit into GPU memory for larger batch sizes.

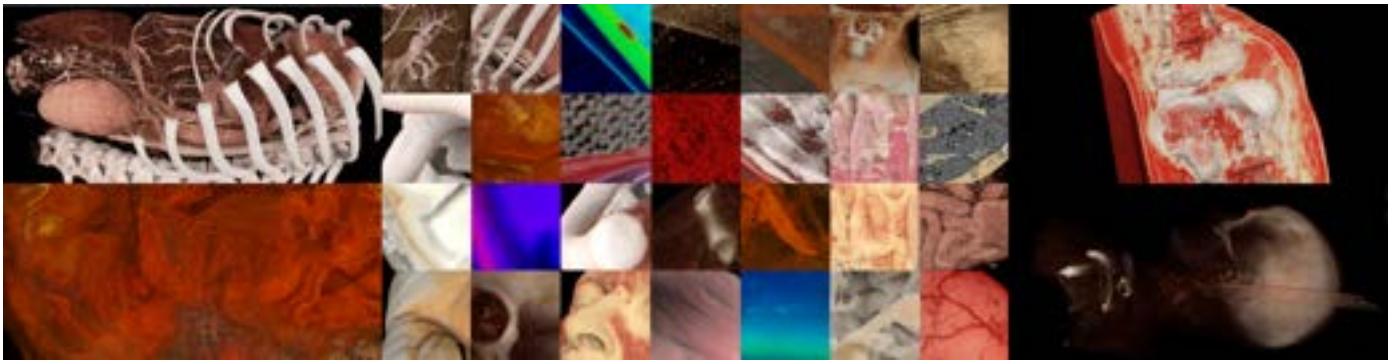

Figure 5. Random images extracted from training data generated by AR^{2}T [4] from CBCT, spiral CT, and MRI scans, until convergence.

The generation of the training dataset comprising 63751 images took about 60 h, while the training process for a single model took approximately 20 h. After training all models for denoising 10 SPP images, we visually analyzed the quality and speed of denoising. The only models that we could run in an acceptable time on GPU during interactions for a viewport size of 1920 × 1080 were AE Lite and AE Full autoencoders, and OIDN. However, AE Full demonstrated better denoising quality on 10 SPP, so AE Lite was excluded from the training on 1 SPP images. Thus, the training for 1 SPP input was conducted only for AE Full and OIDN. The training took a total of 5 days for five 10 SPP models and two 1 SPP models.

4.2. Denoising Timing

The denoising speed and the rendering performance for the image of size 1920 × 1080 is presented in Table 1. For denoising, the timing reported in Table 1 includes the splitting (2 ms) and reconstruction (3 ms) steps for Samsung, Tyan, and OIDN models.

Table 1. Average time cost (ms) of each denoising and performance in frames-per-second (FPS) for rendering and denoising at 1920 × 1080 resolution on an Intel i5-7600K CPU @ 3.80 GHz, 16 GB RAM, and NVIDIA GTX 1060 6 GB. FPS reflects denoising on the GPU wherever possible, except for OIDN STD* and OIDN ADV* ('*' denotes the pre-trained models), which were CPU-denoised due to Pascal architecture incompatibility with the OIDN C++ library.

Model	CPU (ms)	GPU (ms)	1 SPP (FPS)	10 SPP (FPS)
AE Lite	1510	73	7.14	1.35
AE Full	1969	89	6.41	1.32
Samsung	28,374	1108	0.85	0.56
Tyan	28,214	1148	0.82	0.55
OIDN	6640	169	4.24	1.19
OIDN STD*	914	N/A	1.02	0.63
OIDN ADV*	956	N/A	0.98	0.62

4.3. Results Evaluation

The evaluation was performed on 220 images of CBCT scans not included in the training set, separately for 1 SPP and 10 SPP inputs. Among these, 200 images were generated with transfer functions and light configurations used for training, while the other 20 used parameters unseen during training. Therefore, the evaluation was conducted separately for three groups of images. In the evaluation, our primary focus was on the results with trained parameters, as users rarely change them in practice. However, for robustness testing, a limited number of images with unseen parameters were also included in the evaluation.

Figures 6–8 show the results of denoising 1 SPP and 10 SPP images with different characteristics: mostly surface (Figure 6), mostly isotropic (Figure 8), and mixed (Figure 7) test images. Figure 9 provides the quantitative evaluation of different models using SSIM for images with training parameters and using transfer functions and light positions not seen during training. Figures A1 and A2 of Appendix A demonstrate LDR-ꟻLIP and PSNR. The values of the mean and standard deviation are reported in Tables 2 and 3.

As shown in Tables 2 and 3, models trained from scratch achieve higher PSNR and SSIM values compared to pre-trained models (a similar trend is observed for LDR-ꟻLIꟻ, where smaller values indicate better results). Among the models, OIDN and Tyan consistently lead. The only exception is the PSNR for images generated with light position unseen during training, where the PSNR value of OIDN STD* is slightly higher.

While it is generally understood that retraining models on domain-specific data can improve performance, our study demonstrates a less obvious insight: training a model from scratch on a limited quantity of domain-specific data can outperform pre-trained models trained on larger, more diverse datasets. This finding challenges the conventional reliance on pre-trained models, especially in scenarios where the domain-specific nature of the data is fundamental to achieving optimal performance.

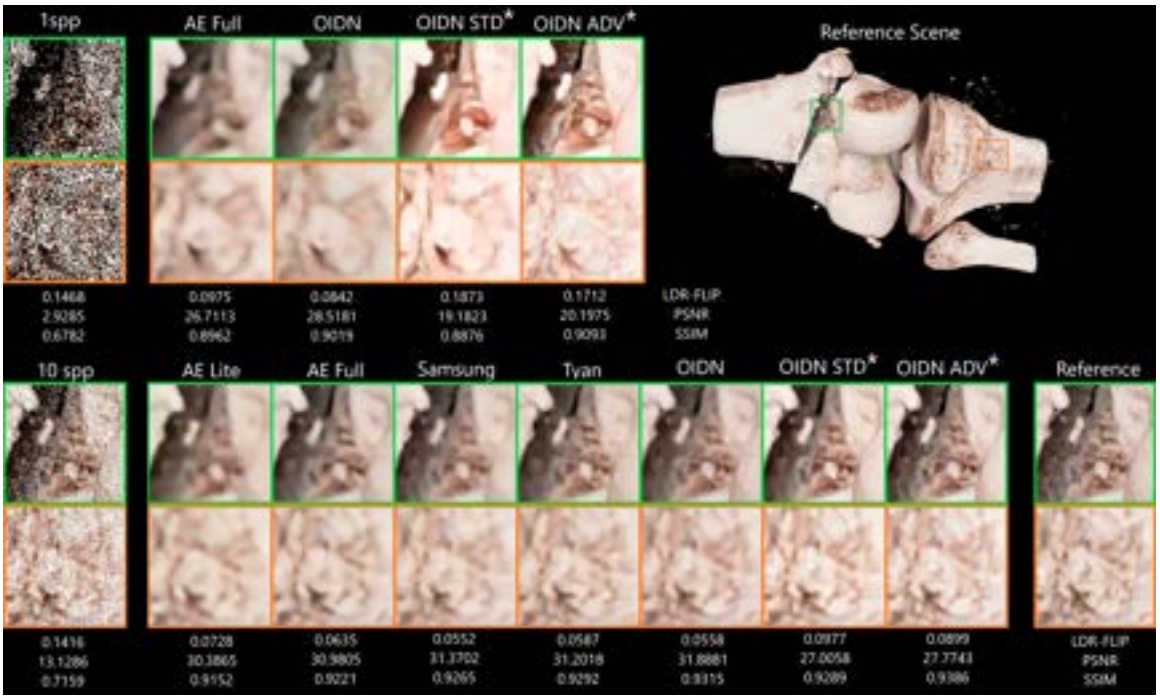

Figure 6. Visual comparison of denoising single-sample-per-pixel (1 SPP) and 10 SPP inputs for a scene predominantly composed of surfaces. The reference scene was generated from a CBCT scan of a broken knee (post-mortem) with 659 SPP. The **upper row** shows two crops, multiplied by 4, of denoising results using 4 different models with 1 SPP input, while the **lower row** displays results from 7 models with 10 SPP input. '*' indicates pre-trained models. Colored squares mark the cropped regions. The metrics reported under each model were calculated on entire image.

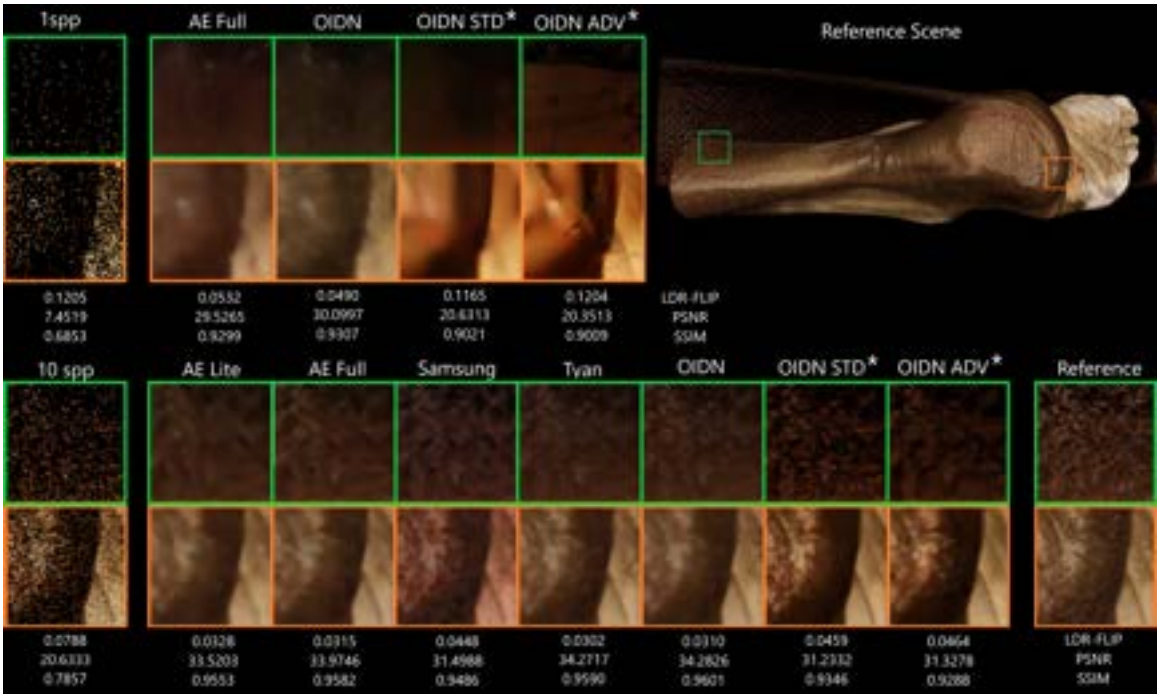

Figure 7. Visual comparison of denoising 1 SPP and 10 SPP inputs for a scene composed of surfaces and semi-transparent volumetric participating media. The reference scene was generated from a CBCT scan of a human leg (post-mortem) with 1083 SPP.

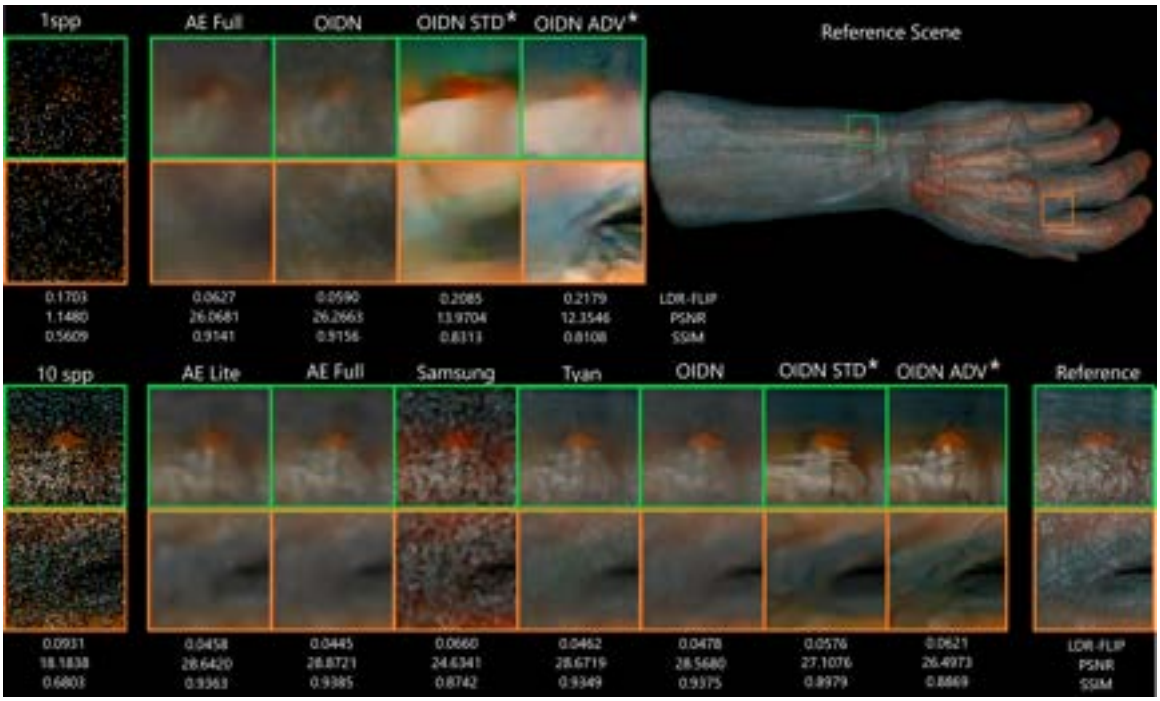

Figure 8. Visual comparison of denoising 1 SPP and 10 SPP inputs for a scene predominantly composed of semi-transparent volumetric participating media. The reference scene was generated from a CBCT scan of a human wrist (post-mortem) with 1675 SPP.

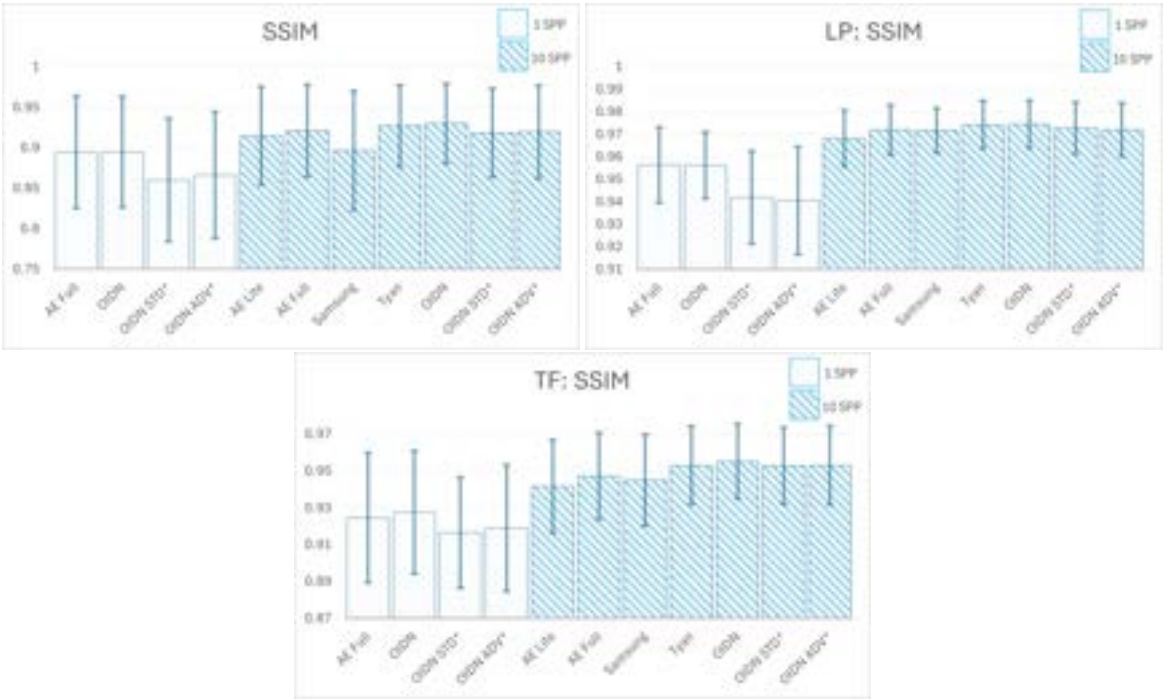

Figure 9. Quantitative comparison of denoising using structural similarity index (SSIM, higher is better), evaluated on: 200 CBCT scan images not included in the training set, generated using transfer functions and light positions seen during the training (**top-left**); 10 images generated using light position, unseen during the training (**top-right**); 10 images generated using transfer functions, unseen during the training (**bottom**). The standard deviation is represented by blue error bars, while dotted bars correspond to models run on 1 SPP input, and dashed bars correspond to models run on 10 SPP input; '*' indicates pre-trained models.

Table 2. Mean and standard deviation (SD) of LDR-ꟻLIP (lower is better), PSNR (higher is better), and SSIM (higher is better) illustrated in Figures A1 and A2 of Appendix A, and Figure 9, for 1 SPP. 'Known TF, LP' refers to transfer function and light position used during training, 'Unknown LP' refers to light position unseen during training, and 'Unknown TF' refers to transfer function unseen during training. The best values are indicated in bold.

Model	Known TF, LP	Unknown LP	Unknown TF
		LDR-ꟻLIP (1 SPP)	
AE Full	0.1022 (0.0657)	0.0413 (0.0156)	0.0787 (0.0407)
OIDN	**0.1020 (0.0685)**	**0.0401 (0.0143)**	**0.0740 (0.0355)**
OIDN STD*	0.2572 (0.1448)	0.0729 (0.0285)	0.1584 (0.0656)
OIDN ADV*	0.2685 (0.1604)	0.0693 (0.0284)	0.1551 (0.0672)
		PSNR (1 SPP)	
AE Full	28.00 (3.17)	**30.15 (3.12)**	29.08 (4.31)
OIDN	**28.41 (2.82)**	30.05 (2.14)	**29.65 (3.28)**
OIDN STD*	19.02 (4.13)	24.28 (2.99)	21.61 (2.78)
OIDN ADV*	18.78 (4.87)	24.89 (3.03)	22.09 (3.04)
		SSIM (1 SPP)	
AE Full	0.8944 (0.0690)	**0.9564 (0.0168)**	0.9245 (0.0353)
OIDN	**0.8945 (0.0685)**	0.9562 (0.0148)	**0.9275 (0.0334)**
OIDN STD*	0.8598 (0.0761)	0.9419 (0.0207)	0.9164 (0.0301)
OIDN ADV*	0.8661 (0.0784)	0.9405 (0.0239)	0.9189 (0.0343)

Table 3. Mean and standard deviation (SD) of LDR-ꟻLIP, PSNR, and SSIM, illustrated in Figures A1 and A2 of Appendix A, and Figure 9, for 10 SPP.

Model	Known TF, LP	Unknown LP	Unknown TF
		LDR-ꟻLIP (10 SPP)	
AE Lite	0.0739 (0.0506)	0.0299 (0.0115)	0.0510 (0.0237)
AE Full	0.0733 (0.0459)	0.0289 (0.0107)	0.0499 (0.0217)
Samsung	0.0963 (0.0801)	0.0262 (0.0091)	0.0600 (0.0341)
Tyan	**0.0625 (0.0398)**	**0.0248 (0.0107)**	0.0423 (0.0174)
OIDN	0.0639 (0.0413)	0.0295 (0.0145)	**0.0419 (0.0174)**
OIDN STD*	0.1049 (0.0668)	0.0265 (0.0104)	0.0700 (0.0333)
OIDN ADV*	0.1064 (0.0729)	0.0268 (0.0106)	0.0698 (0.0356)
		PSNR (10 SPP)	
AE Lite	31.04 (2.98)	32.58 (3.53)	32.42 (2.67)
AE Full	31.33 (2.92)	32.77 (3.70)	32.99 (2.59)
Samsung	29.68 (3.43)	34.11 (2.40)	32.37 (2.77)
Tyan	32.15 (3.00)	34.20 (4.00)	33.90 (2.86)
OIDN	**32.29 (3.05)**	33.08 (4.71)	**34.29 (2.75)**
OIDN STD*	29.07 (3.42)	**34.28 (2.85)**	30.98 (3.54)
OIDN ADV*	29.27 (3.63)	34.27 (2.97)	31.24 (3.57)
		SSIM (10 SPP)	
AE Lite	0.9146 (0.0608)	0.9681 (0.0125)	0.9415 (0.0254)
AE Full	0.9210 (0.0567)	0.9719 (0.0110)	0.9472 (0.0235)
Samsung	0.8964 (0.0736)	0.9716 (0.0100)	0.9449 (0.0248)
Tyan	0.9269 (0.0506)	0.9742 (0.0106)	0.9530 (0.0213)
OIDN	**0.9299 (0.0492)**	**0.9744 (0.0105)**	**0.9551 (0.0203)**
OIDN STD*	0.9183 (0.0549)	0.9727 (0.0115)	0.9527 (0.0209)
OIDN ADV*	0.9196 (0.0574)	0.9719 (0.0119)	0.9530 (0.0215)

4.4. Clinical Assessment

Three datasets were selected for evaluation: a dog with kidney calcifications, a foot with a tiny splinter, and a skull with metal plate insertion (see Figure 10). The survey is available on SurveyMonkey [44]. Fifty-one medical experts evaluated images generated with 1 SPP and 10 SPP and subsequently denoised, as described in Section 3.8. Figure 11 reports the survey results aimed at reflecting the visual assessment performed by medical experts.

According to the survey, denoising applied during interactions and upon mouse release is preferable to Gaussian filter application. However, denoising applied during interactions needs improvement, while results upon mouse release are satisfactory. The assessment results achieved with OIDN ADV* are noteworthy and superior to OIDN, in contrast with the quantitative results of PSNR, SSIM, and LDR-ꟻLIP. Upon analyzing the results, particularly the responses to open-ended questions, we observed that OIDN ADV* received higher subjective ratings due to perceptions such as: "Although it appears too 'plastic', the image processed by AI looks 'cleaner' than the original and, in some ways, better than the reference by enhancing the bone margins". In other words, clinicians sometimes preferred these "false" images ("better than the reference") because of additional qualities such as brightness, contrast, and sharpness. In a subsequent study with an extended survey of 74 participants (presented in a separate publication [45]), we analyzed the survey results in greater detail. This complementary work provides a deeper perspective beyond the scope of the current manuscript.

These findings highlight the need for objective metrics that better correlate with qualitative evaluations and the importance of ensuring that denoising methods meet clinical standards.

Figure 10. Three datasets selected for the survey: (**a**) Cone Beam CT (CBCT) 3D reconstruction of the abdomen and pelvis of a small dog; this reconstruction provides clear visualization of intestinal loops, abdominopelvic vessels, and the left hip, particularly highlighting the ischium, pubis, and coxofemoral joint, (**b**) CBCT 3D reconstruction of the normal human hindfoot with a tiny splinter; the scan offers detailed visualization of key anatomical structures including the calcaneus, talus, tibiotalar joint, and tibiofibular joint, and (**c**) CBCT 3D reconstruction of a human skull in a patient with a fixed orthodontic appliance, undergoing surgical intervention for a Le Fort type I fracture; notable features include the horizontal fracture above the apical roots of the upper dental arch, along with a fracture of the left mandibular angle.

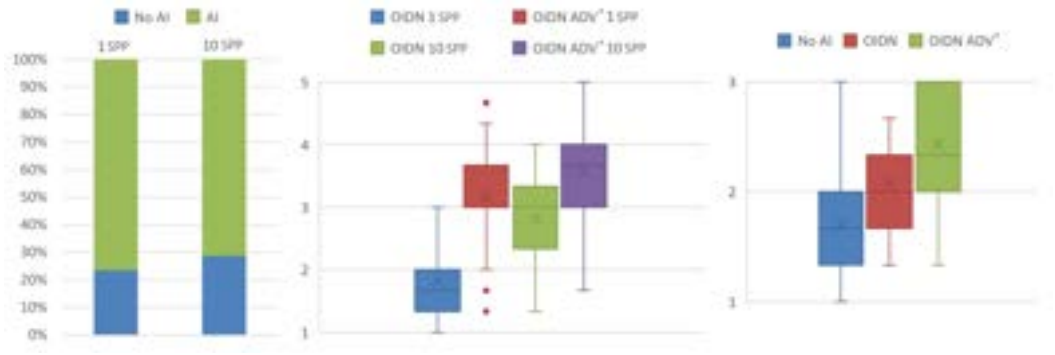

Figure 11. Assessment survey results based on three datasets represented in Figure 10. Fifty-one medical experts participated in the survey. **Left**: Preferences between OIDN denoised images ('AI') and images filtered with Gaussian blur ('No AI') during interactions ('1 SPP') and upon mouse release ('10 SPP'). **Middle**: Boxplots representing participants' confidence levels in images denoised with OIDN and OIDN ADV* during interactions and upon mouse release (rated from 1—*Not confident at all* to 5—*Extremely confident*). **Right**: Evaluation of video previews of interactions with the datasets (rated from 1—*I can't distinguish anything* to 3—*I can distinguish a lot of details*). Crosses in the boxplots represent the mean values of the data, while the boxes show the interquartile range (IQR) with the line inside indicating the median. Whiskers represent the range of non-outlier data, and dots indicate outliers.

4.5. Temporal Stability

Although the focus of this study was not on achieving temporal stability, we estimated it using two denoisers that demonstrated the best quantitative scores. Figure 12 illustrates the temporal stability evaluated with tPSNR on sequences generated from three datasets used for clinical evaluation. The lower peaks correspond to transitions between 1 SPP and 10 SPP, such as when the mouse was released. OIDN trained from scratch consistently yielded higher results, but it is notable that the peaks for the 'Dog' dataset are deeper compared to OIDN ADV*. The videos generated from these sequences can be found in the Supplementary Materials.

Figure 12. Temporal PSNR (tPSNR) evaluated on videos generated from datasets presented in Figure 10. Two sequences were analyzed per dataset: one denoised with OIDN and the other with OIDN ADV*.

5. Limitations and Future Work

As mentioned in Section 3.2, the denoising time must align with data interaction requirements. According to Table 1, the denoising speed using the AE Full model for 1 SPP is 89 ms, reducing the rendering speed on average from 15 FPS to 6 FPS on an NVIDIA GTX 1060. This performance drop could hinder practical usability on hardware with limited capabilities. Furthermore, since the framework used in this study implements the rendering algorithm in OpenGL, the memory transfer between the CPU and GPU during the denoising process introduces additional overhead. Therefore, aside from utilizing more

capable hardware, adopting APIs like Vulkan could be preferable, as they provide direct access to GPU memory, reducing such bottlenecks. To assess the practical performance, we tested our framework on NVIDIA RTX A4000 using CUDA–OpenGL interoperability to minimize CPU–GPU data transfers. With these optimizations, the system achieved an interactive frame rate of 15 FPS (against 24 FPS without deep learning denoising), which is suitable for clinical use. Profiling revealed that the majority of the processing time is spent in denoising inference (27 ms for AE Full), while CPU–GPU transfer overhead was significantly reduced by the CUDA–OpenGL pipeline (about 1 ms). These results demonstrate that, on modern hardware, our framework can maintain real-time interaction while delivering deep learning denoising.

Analyzing the results in Figures 6–8 and Tables 2 and 3, alongside the observed interaction speed degradation, we conclude that the chosen models do not offer significant advantages over Gaussian blur, especially given the current hardware configuration. According to the survey, images generated during interactions were evaluated as too blurry and insufficient for diagnostic purposes—particularly for detecting bone pathologies or fractures—although vessels could be reliably assessed using OIDN ADV*. For images upon mouse release, OIDN ADV* received higher evaluations. It was noted that bones and joints can be seen clearly; however, metallic implants and the surrounding bone tissue are still blurred and not well-defined. This suggests that focal bone lesions or fractures could be identified with high confidence, but evaluating complications (e.g., fracture with consolidation or loosening of the prosthesis or bone infections) around inserted metallic material post-fracture (e.g., nails, plates, synthesis means) or post-arthrosis (e.g., knee or hip prostheses) requires further improvement.

Given the positive evaluation of OIDN ADV* by medical experts, future work on denoising quality improvement should consider the use of HDR training and G-buffers. In the current study, our model trained from scratch used LDR targets, which were sufficient to achieve higher quantitative metrics. However, this choice may have influenced perceptual qualities valued by clinicians, such as natural brightness, contrast, and sharpness. Training with HDR images could potentially provide richer intensity information and improve perceptual realism, offering valuable insights into pre-processing choices and better alignment between quantitative metrics and subjective evaluations. Furthermore, the inclusion of an additional structural guidance map could support improved reconstruction by providing auxiliary information about edges and tissue properties, potentially enhancing denoising quality in both quantitative and qualitative terms.

Temporal denoising remains a promising direction for future work, particularly for addressing frame-to-frame consistency in interactive rendering. Additionally, transformer-based architectures, such as Restormer by Zamir et al. [46], which represent the state-of-the-art in natural image denoising, should be explored for application to MCPT images, particularly in the domain of biomedical volumetric media.

In this work, we did not employ mixed-precision inference, and all models were executed in full FP32 precision. Mixed-precision (FP16/FP32) inference could reduce memory consumption and improve inference speed by exploiting GPU tensor cores, particularly for larger models such as Samsung/Tyan. This approach was beyond the scope of the current study but could be further developed in future work.

Training and evaluation of the denoising models presented in this paper were limited to preset light configurations and transfer function settings, as the current visualization framework does not allow modifying the lighting, and users typically rely on the preset transfer function parameters. This limitation may restrict assessment of generalization to unseen scenarios and should be taken into account.

One limitation of this study lies in the clinical evaluation methodology. While the survey with over 50 medical experts provided valuable insights, the assessment could be enhanced by introducing more structured and objective criteria. For instance, future studies could focus on selecting specific anatomical landmarks, such as the vertebral body, the anterior or posterior arch of a rib, or the lumen of the ascending colon. A precise scoring system could then be developed, employing a 5-point scale to evaluate the quality of these specific landmarks. This approach would make the evaluation more consistent and objective, offering a standardized framework for comparing denoising performance across different models. Future work could also explore automating this evaluation using specifically designed neural networks or consider assessing the overall image quality as perceived by physicians.

6. Conclusions

This study presents several contributions to the field of denoising in biomedical imaging. First, it demonstrates the effectiveness of training deep learning models from scratch using limited domain-specific volumetric data, which outperforms pre-trained models and challenges the reliance on large, diverse datasets. Second, it highlights the disconnect between traditional quantitative metrics and the subjective evaluations of physicians, emphasizing the need for objective metrics that better align with clinical relevance. Additionally, the study proposes a GPU-efficient, tile-based denoising method to overcome memory constraints when processing large medical volumes, enabling practical applications on hardware with limited capabilities. A comprehensive evaluation framework is also provided, integrating traditional metrics such as PSNR, SSIM, LDR-ꟻLIP, and tPSNR with subjective clinical assessments to deliver a well-rounded perspective on denoising quality. These contributions collectively advance the understanding and application of denoising in biomedical imaging, paving the way for future innovations in the field.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app15189893/s1. S1–S2 (Dog): OIDN (S1) and OIDN ADV (S2) denoised sequences. S3–S4 (Foot): OIDN (S3) and OIDN ADV (S4) denoised sequences. S5–S6 (Skull): OIDN (S5) and OIDN ADV (S6) denoised sequences. Videos show temporal stability of the respective denoising methods.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, resources, data curation, writing—original draft preparation, writing—review and editing, E.D.; data curation, visualization, C.N.; supervision, L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The training code, pre-trained domain-specific models, as well as inference script have been made publicly available via Google Drive https://drive.google.com/file/d/18oFsC4HWrnMYs3umpNoVEWs10YPe_JxG/view?usp=drive_link (accessed on 7 September 2025). The training data are also available via Google Drive https://drive.google.com/file/d/1BF72xx3TLZPQ6EUHyyKtLX045PuOORsG/view?usp=drive_link (accessed on 7 September 2025).

Acknowledgments: The authors would like to thank Eleonora Tosca, for her help in selecting the datasets for the survey; Craig Glaiberman, at Epica International, for his invaluable assistance in survey design; Tommaso Paoli, from Santa Maria Annunziata Hospital, Department of Orthopaedic Surgery, for participating in the survey and assisting with participant recruitment. The research is partially supported by Imaginalis s.r.l. During the preparation of this paper, the authors used

ChatGPT-4-turbo to improve English grammar and readability. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the paper.

Conflicts of Interest: Author Elena Denisova was employed by the Imaginalis s.r.l. The remaining author declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LDR-ꟻLIP	Low dynamic range foveated learned image perceptual
CBCT	Cone beam computed tomography
MCPT	Monte Carlo path tracing
OIDN	Open image denoise
PSNR	Peak signal-to-noise ratio
tPSNR	Temporal peak signal-to-noise ratio
SSIM	Structural similarity index measure
CPU	Central processing unit
GPU	Graphics processing unit
HDR	High dynamic range
MRI	Magnetic resonance imaging
MSE	Mean squared error
SPP	Sample(s) per second
AI	Artificial intelligence
CI	Confidence Interval
CT	Computed tomography
kV	Kilovolt
mA	Milliampere
ms	Millisecond
3D	Three-dimensional

Appendix A

Table A1. Comparison of model architectures and key features.

Architecture	Type	Layers	Parameters	Key Features
AE Lite	Autoencoder	Conv2D, MaxPooling2D, UpSampling2D, BatchNorm.	52,451	Lightweight architecture with minimal layers in encoder/decoder **Pros**: Fast training, low resource requirements **Cons**: May lack precision for complex denoising tasks in MCPT
AE Full	Autoencoder	Conv2D, MaxPooling2D, UpSampling2D, BatchNorm.	188,675	Larger model with deeper structure and repeated Conv2D layers **Pros**: Better performance on complex data **Cons**: Higher computational cost and risk of overfitting
Samsung	U-Net	Conv2D, Activation, Add, Concatenate	683,907	Multi-scale residual block, feature fusion, atrous spatial pyramid pooling (ASPP), dilated convolutions **Pros**: Advanced feature fusion, effective for complex noise patterns **Cons**: Computationally intensive, slow inference for real-time denoising
Tyan	U-Net	Conv2D, MaxPooling2D, UpSampling2D, Add, Concatenate	10,918,915	Deep encoding/decoding with dilated convolutions **Pros**: High accuracy for deep denoising tasks **Cons**: Long training times and large memory usage
OIDN	U-Net	Conv2D, MaxPooling2D, UpSampling2D, Add, Concatenate	914,627	Efficient upsampling with feature concatenation in multiple encoder/decoder blocks **Pros**: Efficient upsampling, suitable for real-time applications **Cons**: Limited in handling highly noisy datasets

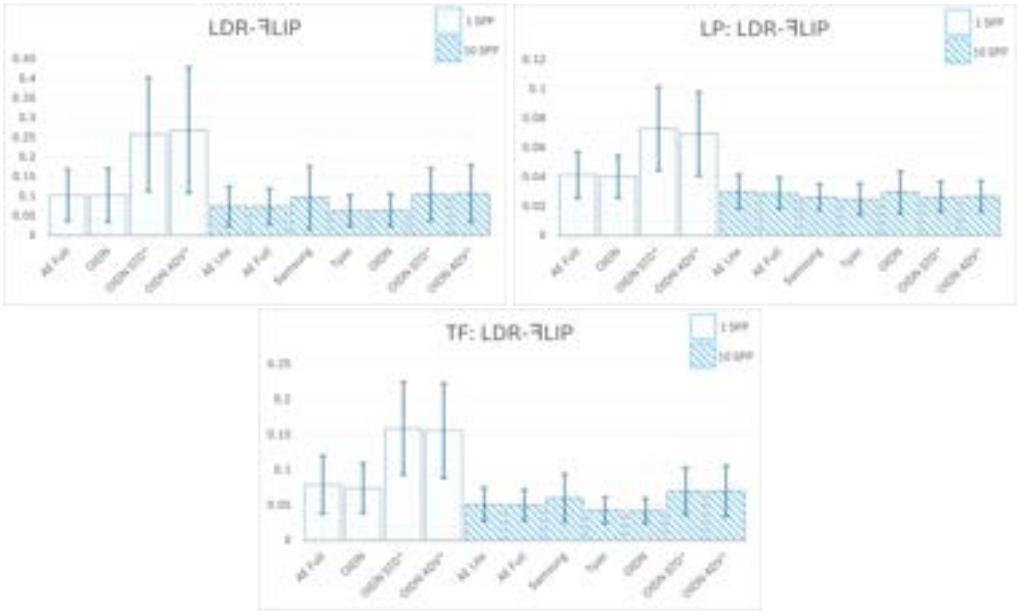

Figure A1. Quantitative comparison of denoising using LDR-FLIP (lower is better), evaluated on: 200 CBCT scan images not included in the training set, generated using transfer functions and light positions seen during the training (**top-left**); 10 images generated using light position, unseen during the training (**top-right**); 10 images generated using transfer functions, unseen during the training (**bottom**). The standard deviation is represented by blue error bars, while dotted bars correspond to models run on 1 SPP input, and dashed bars correspond to models run on 10 SPP input; '*' indicates pre-trained models.

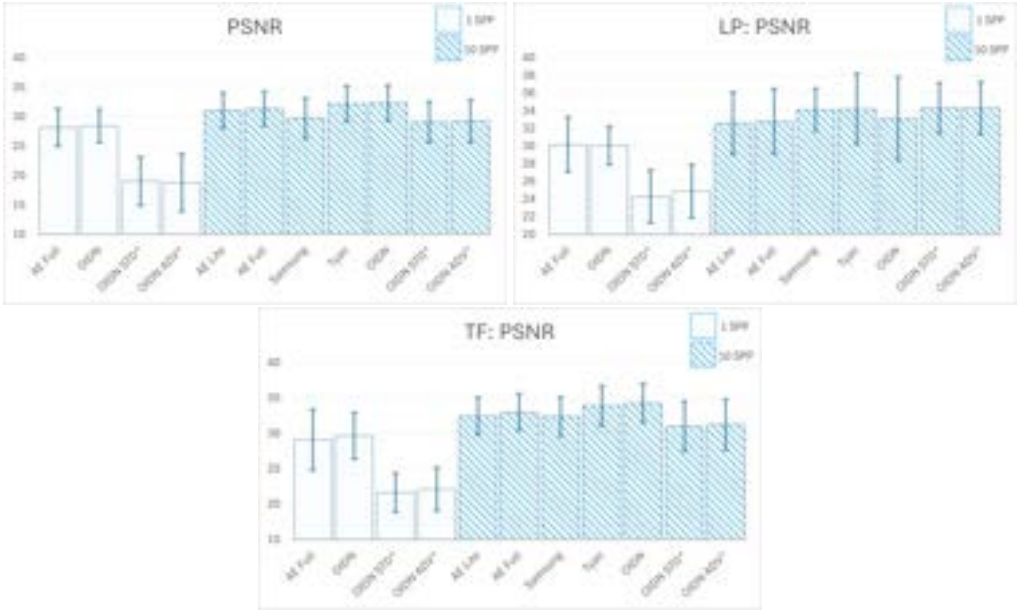

Figure A2. Quantitative comparison of denoising using PSNR (higher is better), evaluated on: 200 CBCT scan images not included in the training set, generated using transfer functions and light positions seen during the training (**top-left**); 10 images generated using light position, unseen during the training (**rop-right**); 10 images generated using transfer functions, unseen during the training (**bottom**).

References

1. Bueno, M.R.; Estrela, C.; Granjeiro, J.M.; Estrela, M.R.d.A.; Azevedo, B.C.; Diogenes, A. Cone-beam computed tomography cinematic rendering: Clinical, teaching and research applications. *Braz. Oral Res.* **2021**, *35*, e024. [CrossRef]
2. Ebert, L.C.; Schweitzer, W.; Gascho, D.; Ruder, T.D.; Flach, P.M.; Thali, M.J.; Ampanozi, G. Forensic 3D visualization of CT data using cinematic volume rendering: A preliminary study. *Am. J. Roentgenol.* **2017**, *208*, 233–240. [CrossRef]
3. Dappa, E.; Higashigaito, K.; Fornaro, J.; Leschka, S.; Wildermuth, S.; Alkadhi, H. Cinematic rendering—An alternative to volume rendering for 3D computed tomography imaging. *Insights Imaging* **2016**, *7*, 849–856. [CrossRef]
4. Denisova, E.; Manetti, L.; Bocchi, L.; Iadanza, E. AR2T: Advanced Realistic Rendering Technique for Biomedical Volumes. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2023: 26th International Conference, Vancouver, BC, Canada, 8–12 October 2023; Proceedings, Part VI; Springer: Berlin/Heidelberg, Germany, 2023; pp. 347–357. [CrossRef]
5. Kajiya, J.T. The Rendering Equation. In Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 31 August 1986; Association for Computing Machinery: New York, NY, USA, 1986; SIGGRAPH '86. pp. 143–150. [CrossRef]
6. Hofmann, N.; Hasselgren, J.; Munkberg, J. Joint Neural Denoising of Surfaces and Volumes. *Proc. ACM Comput. Graph. Interact. Tech.* **2023**, *6*, 10. [CrossRef]
7. Andersson, P.; Nilsson, J.; Akenine-Möller, T.; Oskarsson, M.; Åström, K.; Fairchild, M.D. FLIP: A Difference Evaluator for Alternating Images. *Proc. ACM Comput. Graph. Interact. Tech.* **2020**, *3*, 15. [CrossRef]
8. Hasselgren, J.; Munkberg, J.; Salvi, M.; Patney, A.; Lefohn, A. Neural Temporal Adaptive Sampling and Denoising. *Comput. Graph. Forum* **2020**, *39*, 147–155. [CrossRef]
9. Lee, M.E.; Redner, R.A. Filtering: A note on the Use of Nonlinear Filtering in Computer Graphics. *IEEE Comput. Graph. Appl.* **1990**, *10*, 23–29. [CrossRef]
10. Rushmeier, H.E.; Ward, G.J. Energy preserving non-linear filters. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 24 July 1994; Association for Computing Machinery: New York, NY, USA, 1994; SIGGRAPH '94; pp. 131–138. [CrossRef]
11. Peng, J.; Strela, V.; Zorin, D. A simple algorithm for surface denoising. In Proceedings of the Conference on Visualization '01, San Diego, CA, USA, 21–26 October 2001; VIS '01; pp. 107–112. [CrossRef]
12. Xu, R.; Pattanaik, S.N. A novel Monte Carlo noise reduction operator. *IEEE Comput. Graph. Appl.* **2005**, *25*, 31–35. [CrossRef]
13. Dammertz, H.; Sewtz, D.; Hanika, J.; Lensch, H.P.A. Edge-avoiding À-Trous wavelet transform for fast global illumination filtering. In Proceedings of the Conference on High Performance Graphics, Saarbrücken, Germany, 25–27 June 2010; HPG '10; pp. 67–75. [CrossRef]
14. Kalantari, N.K.; Sen, P. Removing the noise in Monte Carlo rendering with general image denoising algorithms. *Comput. Graph. Forum* **2013**, *32*, 93–102. [CrossRef]
15. Rousselle, F.; Manzi, M.; Zwicker, M. Robust denoising using feature and color information. *Comput. Graph. Forum* **2013**, *32*, 121–130. [CrossRef]
16. Bu, H.; Xu, Q.; Wu, S.; Guo, Y.; Sbert, M. Detection and Removal for Impulse Noise in Monte Carlo Global Illumination Rendered Images of Highly Glossy Scenes. In Proceedings of the 2015 International Conference on Virtual Reality and Visualization (ICVRV), Xiamen, China, 17–18 October 2015; pp. 125–129. [CrossRef]
17. Bitterli, B.; Rousselle, F.; Moon, B.; Iglesias-Guitián, J.A.; Adler, D.; Mitchell, K.; Jarosz, W.; Novák, J. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. *Comput. Graph. Forum* **2016**, *35*, 107–117. [CrossRef]
18. Boughida, M.; Boubekeur, T. Bayesian Collaborative Denoising for Monte Carlo Rendering. *Comput. Graph. Forum* **2017**, *36*, 137–153. [CrossRef]
19. Kalantari, N.K.; Bako, S.; Sen, P. A machine learning approach for filtering Monte Carlo noise. *ACM Trans. Graph.* **2015**, *34*, 122. [CrossRef]
20. Vogels, T.; Rousselle, F.; Mcwilliams, B.; Röthlin, G.; Harvill, A.; Adler, D.; Meyer, M.; Novák, J. Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph.* **2018**, *37*, 124. [CrossRef]
21. Bako, S.; Vogels, T.; Mcwilliams, B.; Meyer, M.; NováK, J.; Harvill, A.; Sen, P.; Derose, T.; Rousselle, F. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* **2017**, *36*, 97. [CrossRef]
22. Xu, B.; Zhang, J.; Wang, R.; Xu, K.; Yang, Y.L.; Li, C.; Tang, R. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.* **2019**, *38*, 224. [CrossRef]
23. Wong, K.M.; Wong, T.T. Deep residual learning for denoising Monte Carlo renderings. *Comput. Vis. Media* **2019**, *5*, 239–255. [CrossRef]
24. Kettunen, M.; Härkönen, E.; Lehtinen, J. Deep convolutional reconstruction for gradient-domain rendering. *ACM Trans. Graph.* **2019**, *38*, 126. [CrossRef]
25. Huo, Y.; Yoon, S.-e. A survey on deep learning-based Monte Carlo denoising. *Comput. Vis. Media* **2021**, *7*, 169–185. [CrossRef]

26. Hofmann, N.; Martschinke, J.; Engel, K.; Stamminger, M. Neural Denoising for Path Tracing of Medical Volumetric Data. *Proc. ACM Comput. Graph. Interact. Tech.* **2020**, *3*, 13. [CrossRef]
27. Chen, Y.; Lu, Y.; Zhang, X.; Xie, N. Interactive neural cascade denoising for 1-spp Monte Carlo images. *Vis. Comput.* **2023**, *39*, 3197–3210. [CrossRef]
28. Schied, C.; Kaplanyan, A.; Wyman, C.; Patney, A.; Chaitanya, C.R.A.; Burgess, J.; Liu, S.; Dachsbacher, C.; Lefohn, A.; Salvi, M. Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In Proceedings of the High Performance Graphics, Los Angeles, CA, USA, 28–30 July 2017; Association for Computing Machinery: New York, NY, USA, 2017; HPG '17. [CrossRef]
29. Lee, J.; Lee, S.; Yoon, M.; Song, B.C. Real-time monte carlo denoising with adaptive fusion network. *IEEE Access* **2024**, *12*, 29154–29165. [CrossRef]
30. Iglesias-Guitian, J.A.; Mane, P.; Moon, B. Real-Time Denoising of Volumetric Path Tracing for Direct Volume Rendering. *IEEE Trans. Vis. Comput. Graph.* **2020**, *28*, 2734–2747. [CrossRef]
31. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *Trans. Img. Proc.* **2017**, *26*, 3142–3155. [CrossRef]
32. Abdelhamed, A.; Afifi, M.; Timofte, A.; Brown, M.S.; Cao, Y.; Zhang, Z.; Zuo, W.; Zhang, X.; Liu, J.; Chen, W.; et al. NTIRE 2020 Challenge on Real Image Denoising: Dataset, Methods and Results. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2077–2088. [CrossRef]
33. Firmino, A.; Frisvad, J.R.; Jensen, H.W. Progressive denoising of Monte Carlo rendered images. *Comput. Graph. Forum* **2022**, *41*, 1–11. [CrossRef]
34. Áfra, A.T. Intel® Open Image Denoise. 2023. Available online: https://www.openimagedenoise.org (accessed on 7 September 2025).
35. TensorFlow. Available online: https://github.com/tensorflow/tensorflow/releases/tag/v2.14.0 (accessed on 1 August 2023).
36. ONNX. Available online: https://onnx.ai/ (accessed on 1 August 2024).
37. CBCTData. Available online: https://kaggle.com/datasets/imaginar2t/cbctdata (accessed on 20 November 2024).
38. Spiral CT: Manix. Available online: https://public.sethealth.app/manix.raw.gz (accessed on 1 August 2023).
39. MRI: Brain. Available online: https://openneuro.org/datasets/ds001780/versions/1.0.0 (accessed on 20 November 2024).
40. Jain, V.; Seung, H.S. Natural image denoising with convolutional networks. In Proceedings of the 21st International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2008; Curran Associates Inc.: Red Hook, NY, USA, 2008; NIPS'08; pp. 769–776.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
42. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.* **2004**, *13*, 600–612. [CrossRef]
43. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. Multiscale structural similarity for image quality assessment. In Proceedings of the The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, USA, 9–12 November 2003; Volume 2, pp. 1398–1402. [CrossRef]
44. AI-Enhanced Realistic Rendering Technique for Biomedical Volumes. Available online: https://www.surveymonkey.com/r/TLK538B (accessed on 20 November 2024)
45. Denisova, E.; Francia, P.; Nardi, C.; Bocchi, L. Advancements in Biomedical Rendering: A Survey on AI-Based Denoising Techniques. *Comput. Biol. Med.* **2025**, *197 Pt A*, 110979. [CrossRef]
46. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M. Restormer: Efficient Transformer for High-Resolution Image Restoration. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5718–5729. [CrossRef]