



AR²T: Advanced Realistic Rendering Technique for Biomedical Volumes

Elena Denisova^{1,3,4} , Leonardo Manetti^{3,4}, Leonardo Bocchi^{1,4} , and Ernesto Iordanza²

¹ Department of Information Engineering,
University of Florence, 50139 Florence, Italy
elena.denisova@unifi.it

² Department of Medical Biotechnologies, University of Siena, 53100 Siena, Italy

³ Imaginalis S.r.l., 50019 Sesto Fiorentino, Italy

⁴ Eidolab, Florence, Italy

Abstract. Three-dimensional (3D) rendering of biomedical volumes can be used to illustrate the diagnosis to patients, train inexperienced clinicians, or facilitate surgery planning for experts. The most realistic visualization can be achieved by the Monte-Carlo path tracing (MCPT) rendering technique which is based on the physical transport of light. However, this technique applied to biomedical volumes has received relatively little attention, because, naively implemented, it does not allow to interact with the data. In this paper, we present our application of MCPT to the biomedical volume rendering—Advanced Realistic Rendering Technique (AR²T), in an attempt to achieve more realism and increase the level of detail in data representation. The main result of our research is a practical framework that includes different visualization techniques: iso-surface rendering, direct volume rendering (DVR) combined with local and global illumination, maximum intensity projection (MIP), and AR²T. The framework allows interaction with the data in high quality for the deterministic algorithms, and in low quality for the stochastic AR²T. A high-quality AR²T image can be generated on user request; the quality improves in real-time, and the process is stopped automatically on the algorithm convergence, or by user, when the desired quality is achieved. The framework enables direct comparison of different rendering algorithms, i.e., utilizing the same view/light position and transfer functions. It therefore can be used by medical experts for immediate one-to-one visual comparison between different data representations in order to collect feedback about the usefulness of the realistic 3D visualization in clinical environment.

Keywords: Monte-Carlo Path Tracing · Realistic Rendering · Biomedical Volumes Visualization

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43987-2_34.

1 Introduction

Now that the hardware performance has achieved a certain level, 3D rendering of biomedical volumes is becoming very popular, above all, with the younger generation of clinicians that uses to use the forefront technologies in their everyday life. 3D representation has proven useful for faster comprehension of traumas in areas of high anatomic complexity, for surgical planning, simulation, and training [2, 9, 15, 25], [18, 28]. Also, it improves communication with patients, which understand the diagnosis much better if illustrated in 3D [23].

The most popular techniques for volume data rendering are maximum-intensity projection (MIP), iso-surface rendering, and direct volume rendering (DVR). These techniques have their pros and cons, but essentially, they suffer from a lack of *photo realism*.

Our novel Advanced Realistic Rendering Technique (AR²T) is based on Monte-Carlo path tracing (MCPT). Historically, MCPT is thought of as a technique suited to (iso)surfaces rendering [14, 19]. Applied to biomedical volumes, this technique has received relatively little attention, probably because, if naively implemented, it does not allow interaction with the data in real-time [7]. However, due to continuous hardware improvement, the problems that can not be resolved in real-time today, will be resolved in real-time tomorrow. For this reason, we have audaciously decided to apply MCPT to the biomedical volumes in attempt to increases the realism and the level of detail in data representation.

In this paper, we present a practical framework that includes different visualization techniques, including AR²T. Our framework allows the user to interact with the data in high quality for the deterministic algorithms (iso-surface, MIP, DVR), and in low quality for the stochastic AR²T. Moreover, the framework supports a mixed modality that works as follows. By default, the data is rendered by DVR. It allows to interact with the data, adjust the transfer function, and apply clip planes. However, a high-quality AR²T image can be generated at any moment by the user request without explicitly switching between rendering algorithms. The quality improves progressively, and the process can be stopped as soon as the desired quality is achieved. As an alternative, the improvement stops automatically, when the algorithm converged. The framework permits to compare different rendering techniques directly, i.e., using the same view/light position and transfer functions. It, therefore, promotes further research on the importance of realism in visualising biomedical volumes, providing medical experts with an immediate one-to-one visual comparison between different data representations.

Related Work. Various deterministic approaches were applied in an attempt to increase the realism of volume rendering. Above all, the direct volume rendering technique has been enriched with local and global illumination, combined with ambient occlusion and shadowing [13, 21, 24]. However, these approaches are not able to produce photo-realistic images, being based on a very simplified and far-fetched model.

One interesting technique for improved DVR, which includes realistic effects and physically based lighting, was proposed by Kroes et al. in 2012 [17]. They

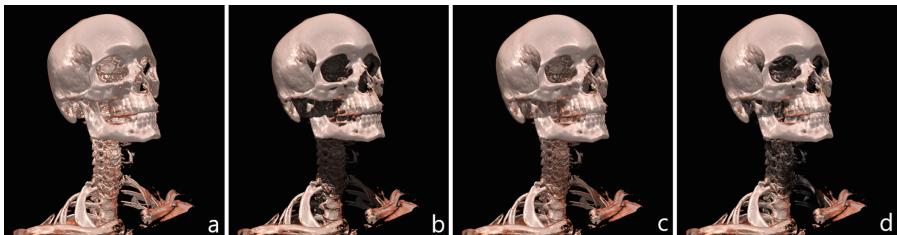


Fig. 1. Direct Volume Rendering. Human skull CBCT rendered with DVR: **a**. Local illumination; **b**. Local and global illumination, linear lighting; **c**. Local and global illumination, conical lighting; **d**. Local and global illumination with translucency, conical lighting. For all techniques, Phong local illumination was used.

were the first who demonstrated that, if properly optimized, ray tracing of volumetric data can be done interactively. However, their method is based on single scattering and consequently does not produce photo-realistic images. Despite that, this approach still arouses interest [5, 30]. In fact, as far as we know, since then no one has presented any different reproducible technique on photo-realistic rendering of biomedical volumetric data.

Recently, the cinematic rendering (CR) 3D technique was introduced [4, 10]. It is available as a part of commercial closed-source software, and the implementation details are not publicly available. Several studies compared DVR and CR images, produced by different software [6–8, 29]. However, the one-to-one comparison is difficult, because it is not practically possible to align the data by means of visual settings and, above all, positioning.

2 Methods

The entire framework is written from scratch in C++ using Qt. The whole rendering runs on GPU and is implemented in OpenGL Shading Language (GLSL). In the following subsections, we describe the techniques we used within the framework, giving the details only for AR²T for the sake of brevity. The quality of the images produced with the different techniques is difficult to assess with a numerical index; following [4, 6–8, 29], we carried out a survey, based on the visual comparison of the proposed methods (see *Results*).

2.1 Deterministic Rendering Algorithms

First of all, we implemented the most popular rendering techniques for biomedical volumetric data: iso-surface, MIP, and DVR. They gave us the basis for the direct comparison of proposed methods. Then, we enriched our DVR model with local and global illumination, applying various approaches [11, 16, 20] in an attempt to improve realism and receive feedback from clinicians (see Fig. 1). It was immediately clear that despite these techniques can improve realism by introducing deep shadows, they are not suitable for the visualization of biomedical volumes because hide information in the shadowed areas without increasing anyhow the level of detail.

2.2 Advanced Realistic Rendering Technique

There are two modalities of AR²T visualization: pure AR²T and mixed DVR-AR²T. When pure AR²T is active, the interactivity is achieved by the execution of just one iteration of the algorithm. When the interaction is finished (e.g., the mouse button is released), 10 iterations of AR²T are executed. To improve the quality, Gaussian blur filter [1] is applied during the interactions, so the overall image is understandable. On request, the iterative algorithm improves the quality until the user stops the process or the convergence is achieved (see subsection *Convergence*). In mixed modality, the interactions are in DVR, and the AR²T runs on request. When the interaction restarts, the visualization automatically switches to DVR.

Our AR²T is inspired by MCPT applied to analytically generated surfaces and isotropic volumes [12, 26]. In our model, we provide advanced camera settings (aperture, focal distance, see Fig. 2) and support an unlimited number of light sources of any shape. For practical reasons, we limit the number of ray scatters to 10 (in our experiments, we did not see any improvement in realism for a larger number of scatters). In the following subsections, we step-by-step describe the implementation details of AR²T, to allow the reproducibility of our results.

GPU Implementation. To be independent in the choice of hardware to run our framework, we implement AR²T in GLSL. Unfortunately, there are two main issues to resolve for Monte-Carlo path tracing in OpenGL: 1. recursion, and 2. random number generation.

Recursion. As GLSL memory model does not allow for recursive function calls, which are essential for MCPT, we simulated the recursion by exploiting *multiple render targets* feature of modern GPUs. This feature allows the rendering pipeline to render images to multiple render target textures at once. Indeed, the information we need after every scatter of a ray is the resulting colour, the position where the scatter occurred, and the direction in which the ray scatters.

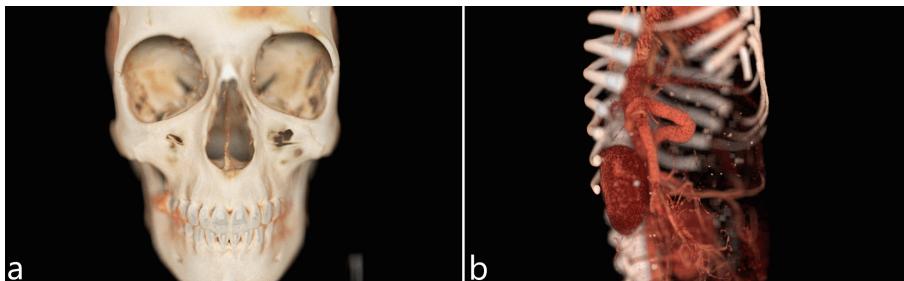


Fig. 2. AR²T: Camera Aperture & Focal Distance. **a.** Human skull CBCT; in focus: nasal bone, infraorbital foramen, vomer. **b.** Dog abdomen CBCT with contrast; in focus: intrahepatic portacaval shunt, main portal vein, right kidney.

Therefore, three target textures are necessary for every rendering step. Moreover, two frame buffers are used in a *ping pong blending* manner (as described in [16]) to enable the reading of textures filled on the previous step. Thus, in the first step, the ray origin and direction are calculated according to the camera properties and position. In the subsequent steps, the ray origin and direction are read from the corresponding textures.

On any step, three situations are possible: (1) The ray does not hit the volume or the light source. In this case, a zero-length vector is saved to *direction* render target - it indicates that the ray scattering finishes here, and the resulting colour components are set to zeros (for ulterior speed up, and considering that the usual background for biomedical visualization is black, we do not model the Cornell box outside the volume). (2) The ray hits the light source. Then, the resulting colour, accumulated until this moment, is attenuated by the light's colour, and, again, the ray scattering finishes here. (3) The volume is hit. The scattering continues, until the ray encounters any of the stopping conditions, or the scatter number limit is achieved. In this case, the resulting colour components are set to zeros.

Random Number Generation. To provide the uniformly distributed random numbers on the fragment stage of the OpenGL pipeline, we generate a pool of 50 additional two-dimensional textures of *viewport* size and fill them with uniformly distributed random numbers generated with **std::uniform_int_distribution**. In each step, we randomly choose four textures from the pool to provide random numbers: two for advanced Woodcock tracking, one for scattering, and one for sampling direction. Every 100 iterations of the algorithm, we regenerate the pool of random numbers to avoid the quality improvement stuck. On Intel(R) Core(TM) i5-7600K CPU @ 3.80 GHz, the generation of the pool takes ~ 1600 ms, for viewport size 1727×822 . It occupies ~ 270 Mb of RAM.

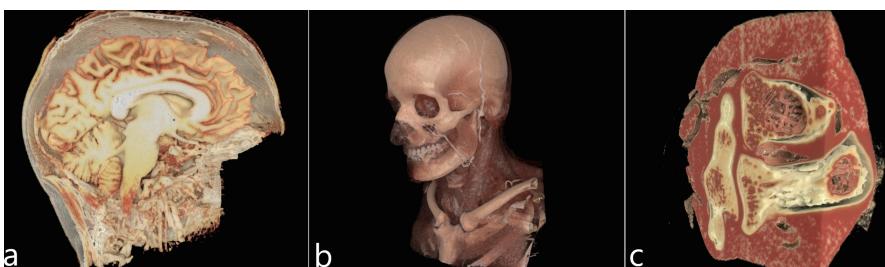


Fig. 3. AR²T: Hard & Soft Tissues. **a.** Human brain MRI, <https://openneuro.org/datasets/ds001780/versions/1.0.0>: identified pre-central gyrus of frontal lobe, genu of the corpus callosum, lateral ventricle, optic nerve, pons, cerebellum, medulla oblongata, cerebellar tonsil; **b.** Spiral CT Manix, <https://public.sethealth.app/manix.raw.gz>; **c.** Human knee CBCT (post-mortem).

Advanced Woodcock Tracking. For volume sampling, we implemented the advanced Woodcock tracking technique, described in [3,27]. When the maximum volume density is much larger than the typical density in the volume, the Woodcock tracking can be improved by breaking the original volume into sub-volumes, each having a small density variation. For this aim, every time the transfer function changes, we construct a voxel grid that has the local maxima of the density in its nodes. Then, the voxel grid is straightforwardly applied on Woodcock tracking, giving up to 5x speed-up respect to the basic implementation. On NVIDIA GeForce GTX 1060, the voxelization process, implemented in GLSL and executed on a fragment stage, takes up to 500 ms for a voxel grid node size $s = 4$, and depending on the volume size. The additional memory needed for the voxel grid storage is $1/s^3$ of the original volume size.

Phase Functions. In AR²T, we use four well-known phase functions, described in [26]: Lambertian, metal, dielectric, and isotropic. Every time the ray hits a volume voxel, we choose which phase function to apply based on the density of the voxel: if the density is less than some threshold t_0 , it causes dielectric scatter; when it is higher than some threshold t_1 , it causes metal scatter; otherwise, as proposed in [26], we randomly decide if to sample toward the light sources or to pick direction according to the voxel reflection (*mixture probability density function*). When we decide to sample according to the hit voxel direction, we choose between surface and volumetric scattering. Following [17], we switch between Lambertian and isotropic phase functions, basing not only on the voxel density but also on the local gradient. Thus, Lambertian is chosen with the following probability:

$$p = 1 - \alpha(\vec{v}) \cdot (1 - e^{-s \cdot m(\vec{v})}), \quad (1)$$

where $\alpha(\vec{v}) \in [t_0, t_1]$ is the voxel density (or opacity), $m(\vec{v})$ is the normalized gradient magnitude, and s is the hybrid scattering factor.



Fig. 4. AR²T: Translucency. **a.** Human foot CBCT (post-mortem); well-distinguished distal tibiofibular joint (syndesmosis), talus, navicular, cuboid, middle cuneiform, fifth metatarsal, under the semitransparent skin. **b.** Cat thorax CBCT; semitransparent lungs and trachea.

Image Generation. When the first iteration of AR²T is completed, the result contained in the colour texture (see *Recursion* for rehearse) is blit into the output rendering frame buffer to be immediately displayed. Moreover, it is saved locally to be summed with the results of the next iterations. On the next iterations, the accumulated result is saved into the local buffer, and then the medium (e.g. the sum divided by the iterations number) is blit into the output frame buffer and displayed.

Convergence. As a convergence criterion, we use mean square displacement (MSD) between the iterations [22]. After each iteration, the square displacement between the current and the previous pixel colour components is calculated directly on a fragment stage. When MSD becomes less than $\epsilon = 5 \cdot 10^{-7}$, the iterations stop, and the method is considered converged (see Fig. 5). In our experiments, the convergence was achieved within 800 iterations for all images, and it took up to 100 s.

3 Results

In our experiments, we have used spiral CT (Computed Tomography), MRI (Magnetic Resonance Imaging), and CBCT (Cone-Beam Computed Tomography) publicly available data sets. The CBCT data sets were acquired by SeeFactorCT3TM (human) and VimagoTMGT30 (vet) Multimodal Medical Imaging Platforms in our layout and are available on <https://kaggle.com/datasets/imaginart/cbctdata>.

To validate the superiority of the AR²T over the other methods implemented in our platform, we ask a group of clinicians to participate into the survey. 22 participants (7 orthopedic surgeons, 1 trauma surgeon, 1 neurosurgeon, 7 interventional radiologists, 6 veterinaries) evaluated the data sets on Fig. 6, visualized

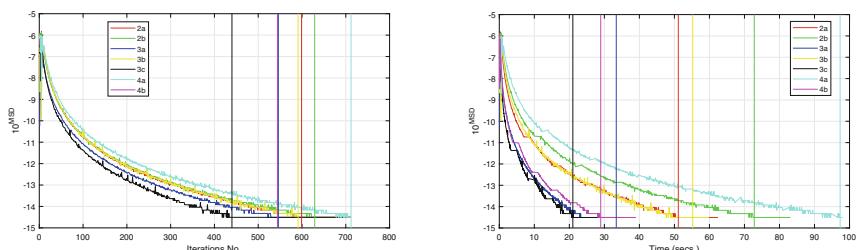


Fig. 5. Convergence Plots (Fig. 2a–4b). Vertical lines indicate the number of iterations needed for convergence (left) and the convergence time in seconds (right). All images were generated on NVIDIA GeForce GTX 1060 6 Gb, Intel(R) Core(TM) i5-7600K CPU @ 3.80 GHz RAM 16 Gb, viewport size 1727 × 822. All data sets are encoded in 16-bit format.

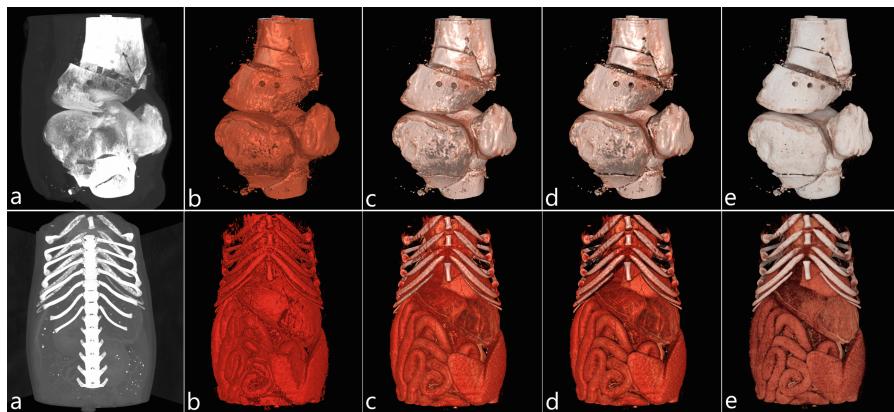


Fig. 6. One-To-One Rendering Comparison. Human knee CBCT post-mortem (top) and dog abdomen CBCT with contrast (bottom): MIP (**a**), Iso-surface (**b**), DVR with local (**c**) and global illumination, linear lighting (**d**), AR²T (**e**). No light/view or transfer function changes across **a–e**.

using MIP, iso-surface, DVR with local illumination, DVR with local and global illumination (linear lighting), and AR²T, voting the best overall image, the most realistic one, the more diagnostic (if any), and the more valuable in their practice.

According to Table 1, the AR²T provides the best overall, the most realistic, diagnostic, and valuable images. Participants commented that the images produced with AR²T “provide better resolution, amazing clarity with less artifacts and noise, better sharpness and contrast, are the closest in colour to real tissue and the most similar to a dissected body deprived of blood, help to understand the anatomy and the pathology of the district, have excellent qualities in general surgery for the definition of the splanchnic organs, for diagnosis and preoperative study”. Meanwhile the others are “either glossy, or too colorized or not as sharp, and seem artificial”. Some participants stated that DVR images are the sharpest and seem to be more detailed (“peritoneal meso is better detected, sub-xiphoid is more visible”), but also it was mentioned that “too much sharpening causes misleading images creating artificial findings”. Some participants noted the diagnostic usefulness of MIP for vascular issues. Participants who stated that none of the images were diagnostic or useful admitted that they did not deal with the presented anatomical structures in their practice or had never used 3D rendering and could not assess its practical application.

Table 1. Survey Responses Indicating How Many of 22 Participants Voted the Method that Produces: Best Overall, Most Realistic, More Diagnostic (If Any), and More Valuable Images, by evaluating Fig. 6

Method	Best Overall		Most Realistic		More Diagnostic		More Valuable	
	Knee	Abs	Knee	Abs	Knee	Abs	Knee	Abs
(e) AR ² T	18	16	17	17	18	12	19	15
(d) DVR [†]	4	3	3	3	1	2	2	3
(c) DVR	—	3	1	2	—	2	—	2
(b) Iso-surf.	—	—	—	—	—	—	—	—
(a) MIP	—	—	1	—	1	—	—	—
None	—	—	—	—	2	6	1	2

Note: *DVR[†]* – local and global illumination, linear lighting; *DVR* – local illumination.

4 Conclusions

The main result of our research is the novel advanced realistic rendering technique AR²T (see Fig. 2, 3, 4), implemented within a practical framework, that allows to compare different rendering techniques directly (Fig. 6).

Despite our model supports any number of light sources, all images (except Fig. 1), presented in this paper, were generated with a single spherical light source, placed right in front of the volume. We plan to dedicate extra time to find the best light configuration from the clinical point of view. Moreover, our future research will be focused on the ulterior improvement of AR²T speed and quality, and the comparison metrics.

At the moment of writing this paper, we are evaluating free access to our framework's executable to share our results, facilitate the comparison with other approaches, and stimulate further research on the usefulness of photo-realistic 3D images in medicine.

References

1. Gaussian blur filter shader. <https://web.archive.org/web/20150320024135/>, <http://www.gamerendering.com/2008/10/11/gaussian-blur-filter-shader/>. Accessed 08 Mar 2023
2. Abou El-Seoud, S., Mady, A., Rashed, E.: An interactive mixed reality ray tracing rendering mobile application of medical data in minimally invasive surgeries (2019)
3. Behlouli, A., Visvikis, D., Bert, J.: Improved woodcock tracking on Monte Carlo simulations for medical applications. Phys. Med. Biol. **63**(22), 225005 (2018). <https://doi.org/10.1088/1361-6560/aae937>
4. 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 Research **35** (2021). <https://doi.org/10.1590/1807-3107bor-2021.vol35.0024>

5. Cheng, H., Xu, C., Wang, J., Chen, Z., Zhao, L.: Fast and accurate illumination estimation using LDR panoramic images for realistic rendering. *IEEE Trans. Visual Comput. Graphics* (2022). <https://doi.org/10.1109/TVCG.2022.3205614>
6. 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* **7**(6), 849–856 (2016). <https://doi.org/10.1007/s13244-016-0518-1>
7. Ebert, L.C., et al.: Forensic 3D visualization of CT data using cinematic volume rendering: a preliminary study. *Am. J. Roentgenol.* **208**(2), 233–240 (2017). <https://doi.org/10.2214/AJR.16.16499>
8. Eid, M., et al.: Cinematic rendering in CT: a novel, lifelike 3D visualization technique. *Am. J. Roentgenol.* **209**(2), 370–379 (2017). <https://doi.org/10.2214/AJR.17.17850>
9. Elshafei, M., et al.: Comparison of cinematic rendering and computed tomography for speed and comprehension of surgical anatomy. *JAMA Surg.* **154**(8), 738–744 (2019). <https://doi.org/10.1001/jamasurg.2019.1168>
10. Engel, K.: Real-time Monte-Carlo path tracing of medical volume data. In: GPU Technology Conference, 4–7 Apr 2016. San Jose Convention Center, CA, USA (2016)
11. Fernando, R., et al.: GPU Gems: Programming Techniques, Tips, and Tricks for Real-time Graphics, vol. 590. Addison-Wesley Reading (2004)
12. Fong, J., Wrenninge, M., Kulla, C., Habel, R.: Production volume rendering: Siggraph 2017 course. In: ACM SIGGRAPH 2017 Courses, pp. 1–79 (2017). <https://doi.org/10.1145/3084873.3084907>
13. Hernell, F., Ljung, P., Ynnerman, A.: Local ambient occlusion in direct volume rendering. *IEEE Trans. Visual Comput. Graphics* **16**(4), 548–559 (2009). <https://doi.org/10.1109/TVCG.2009.45>
14. Jensen, H.W., et al.: Monte Carlo ray tracing. In: ACM SIGGRAPH, vol. 5 (2003)
15. Johnson, P.T., Schneider, R., Lugo-Fagundo, C., Johnson, M.B., Fishman, E.K.: MDCT angiography with 3D rendering: a novel cinematic rendering algorithm for enhanced anatomic detail. *Am. J. Roentgenol.* **209**(2), 309–312 (2017). <https://doi.org/10.2214/AJR.17.17903>
16. Kniss, J., Premoze, S., Hansen, C., Shirley, P., McPherson, A.: A model for volume lighting and modeling. *IEEE Trans. Visual Comput. Graphics* **9**(2), 150–162 (2003). <https://doi.org/10.1109/TVCG.2003.1196003>
17. Kroes, T., Post, F.H., Botha, C.P.: Exposure render: an interactive photo-realistic volume rendering framework. *PloS one* **7**(7), e38596 (2012). <https://doi.org/10.1371/journal.pone.0038586>
18. Kutaish, H., Acker, A., Drittenbass, L., Stern, R., Assal, M.: Computer-assisted surgery and navigation in foot and ankle: state of the art and fields of application. *EFORT Open Rev.* **6**(7), 531–538 (2021). https://doi.org/10.1302/2058-5241.6_200024
19. Lafourture, E.P., Willem, Y.D.: Rendering participating media with bidirectional path tracing. In: Pueyo, X., Schröder, P. (eds.) EGSR 1996. E, pp. 91–100. Springer, Vienna (1996). https://doi.org/10.1007/978-3-7091-7484-5_10
20. Max, N., Chen, M.: Local and global illumination in the volume rendering integral. Technical report, Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States) (2005)
21. McNamara, A.: Illumination in computer graphics. The University of Dublin (2003)

22. Michalet, X.: Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. *Phys. Rev. E* **82**(4), 041914 (2010). <https://doi.org/10.1103/PhysRevE.82.041914>
23. Pachowsky, M.L., et al.: Cinematic rendering in rheumatic diseases-photorealistic depiction of pathologies improves disease understanding for patients. *Front. Med.* **9**, 946106 (2022). <https://doi.org/10.3389/fmed.2022.946106>
24. Salama, C.R.: GPU-based Monte-Carlo volume raycasting. In: 15th Pacific Conference on Computer Graphics and Applications (PG 2007), pp. 411–414. IEEE (2007). <https://doi.org/10.1109/PG.2007.27>
25. Sariali, E., Mauprizez, R., Khiami, F., Pascal-Mousselard, H., Catonné, Y.: Accuracy of the preoperative planning for cementless total hip arthroplasty. a randomised comparison between three-dimensional computerised planning and conventional templating. *Orthop. Traumatol. Surg. Res.* **98**(2), 151–158 (2012). <https://doi.org/10.1016/j.otsr.2011.09.023>
26. Shirley, P., Morley, R.K.: Realistic Ray Tracing. AK Peters Ltd, Natick (2008)
27. Szirmay-Kalos, L., Tóth, B., Magdics, M.: Free path sampling in high resolution inhomogeneous participating media. In: Computer Graphics Forum, vol. 30, pp. 85–97. Wiley Online Library (2011). <https://doi.org/10.1111/j.1467-8659.2010.01831.x>
28. Wang, C., et al.: Patient-specific instrument-assisted minimally invasive internal fixation of calcaneal fracture for rapid and accurate execution of a preoperative plan: a retrospective study. *BMC Musculoskelet. Disord.* **21**, 1–11 (2020). <https://doi.org/10.1186/s12891-020-03439-3>
29. Xu, J., et al.: Interactive, in-browser cinematic volume rendering of medical images. *Comput. Methods Biomed. Eng. Imaging Visual.* **11**, 1–8 (2022). <https://doi.org/10.1080/21681163.2022.2145239>
30. Zhou, S.: Woodcock tracking based fast Monte Carlo direct volume rendering method. *J. Syst. Simul.* **29**(5), 1125–1131 (2017). <https://doi.org/10.16182/j.issn1004731x.joss.201705026>