The Review of Modeling and Rendering 3D Environment: A Survey

Yufei Lin *

Intelligent science and technology, Beijing University of Posts and Telecommunications, Bejing, China

* Corresponding Author Email: 3171426193@qq.com

Abstract. This paper explores several topics in the field of 3D Environments, including ray tracing, rendering techniques, algorithms and optimization, data processing, real-time rendering, modeling, physical simulation, and material modeling. The paper covers a large variety of applications and techniques that are widely used in computer graphics and visualization. This paper also points out that significant progress has been made in ray tracing, texture filtering, image processing, motion synthesis, geometric modeling, and physical simulation. These advances have improved the visual quality, performance, and fidelity of computer-generated images. In addition, the article mentions the trend of integrating machine learning, deep learning, and data analysis techniques into computer graphics and visualization, especially in data processing and analysis models, to improve the results of rendering and simulation. Overall, this article explores details of the current state of the computer graphics and visualization field, highlighting the various technologies and applications which shapes the field.

Keywords: Rendering, Modeling, improved algorithm.

1. Introduction

Rendering refers to the process of generating images from a model by means of computer software. It involves the conversion of a 3D model or scene into a 2D image, taking account of various factors such as lighting, textures, shadows, and so on. Rendering is widely used in fields such as animation, visual effects, architectural visualization, product design, and video games to create realistic or stylized images and animations. The different rendering techniques, including rasterization, ray tracing, and global illumination, each with its own advantages and applications. The choice of rendering technique depends on factors such as the desired visual fidelity, computational resources, and specific requirements of the project. Overall, rendering is a crucial aspect of modern computer graphics and visual communication, allowing artists and designers to transform 3D models into compelling and visually engaging images and animations for a wide range of purposes.

Fig. 1 Moving teapot. These images can be aligned despite the walking teapot moving and perform a noise-based merging [1].

Modeling is a process of creating, describing, and simulating concepts, systems, or phenomena. It plays an important and significant role in science, engineering, art, and many other fields. Modeling helps to understand and explain the complex real world and is used for prediction, optimization, and decision-making. In different fields, modeling can use different methods and techniques. No matter what type of modeling it is, its purpose is to better understand and deal with problems. Modeling helps grasp the essence of complex systems holistically and provides an effective tool for predicting and optimizing system behavior. Through modeling, it can conduct virtual experiments, generate absolutely new insights, and support decision-making and innovation.

With the development of the sciennce, more and more people become intrested in the aplications, games about the knowledge of 3D enviroment. Also, it makes the modeling and rendering in great attraction.

2. Rendering

Rendering is the process of transforming computer-generated graphics or images into the final visual output. In the fields of computer graphics and computer vision it is really important. The rendering process involves multiple stages, including geometry processing, lighting calculations, texture mapping, and projection, among others. Firstly, transforming and cliping the vertices of the 3D model in the stage of geometry processing to determine their positions on the screen. Then, the position, intensity, and color of light sources is considered in the stage of lighting calculation to compute the brightness and color values for each pixel. Next, the texture mapping stage applies material textures to object surfaces to add detail and realism. Finally, the projection stage maps the 3D scene onto a 2D plane to generate the final image or animation.

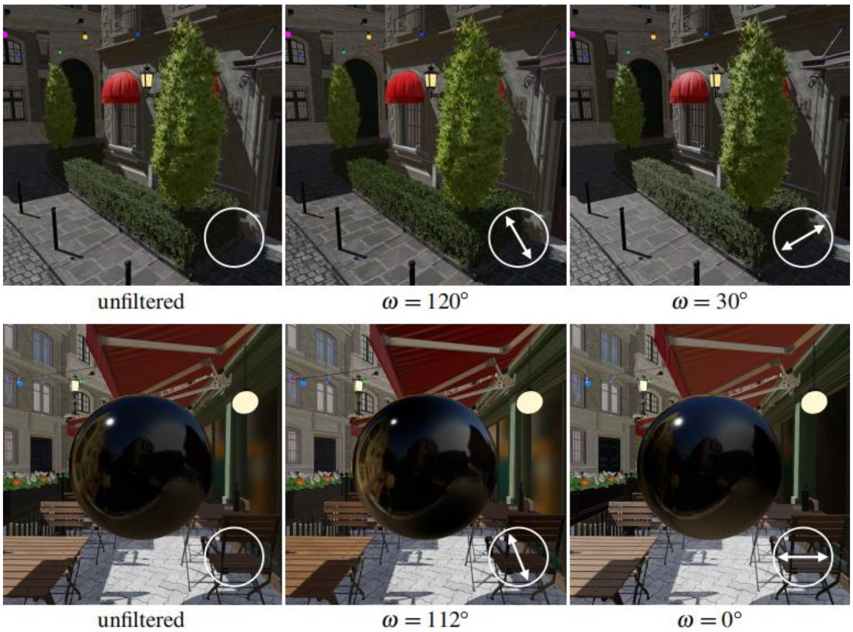

Fig. 2 Two areas of the Bistro Exterior scnce rendered without a polarizing filter (left colum) and with one (middle and rught columns) [2].

2.1. Ray Tracing Related

Ray tracing is an inherent component of modern rendering algorithms. These algorithms model light transport by means of geometric optics, asuming that light travels instantaneously through the medium in straight lines [3]. Rendering solves with integrating multidimensional functions that are usually dealed through numerical integration typically, such as Monte Carlo or quasi-Monte Carlo. It uses a technique named Multidimensional adaptive sampling to reduce error [4]. Then baseing on tracing random ray’s photon mapping is a numerical solution to the rendering equation [5]. Also, ray tracing can compute the intersection between a ray and the surface defined by trilinearly interpolating SDF values from the $2 \times 2 \times 2$ corners of a voxel about signed distances [6]. The paper presents an incremental technique for traversing frustum grids which can be very useful to distribute voxels efficiently in OpenVDB—a widely adopted format for volume storage in production path tracing [7].

Besides, there are many other forms that can be used in ray tracing. Firstly, comparing with traditional spatial search methods through two representative applications it demonstrates the performance, especially on dynamically updated sample sets. These two applications are geometric processing of scanning point clouds and global illumination through progressive photon mapping [8].

Fig. 3 Example renderings that demonstrate our approach in closed enviroments of arbitrary light complexuty.

2.2. Real-time Rendering and Effects

Real-time rendering refers to the process of generating and displaying computer-generated graphics in real-time, typically at interactive frame rates. Real-time shading with free-form planar area lights uses linear transform cosine function to realize real-time rendering [9]. Polarization filter function is a technology in real-time rendering, the influence of polarization filter on rendered images can be used to simulate to enhance the visual quality and realism of real-time rendering scenes [2]. An extension to real-time image-based lighting models that incorporates multiple scattering, and it is suitable for real-time applications on account of its low run-time cost [10]. Then, a lightweight and effective is proposed, which is a method to transform the appearance of materials between different BRDF model and across different renderers, it uses image-based metrics for appearance comparison, and delegates the interaction with the model to the renderer, and it presents an incremental technique for traversing frustum grids in OpenVDB—a widely adopted format for volume storage in production path tracing [7,11,12].

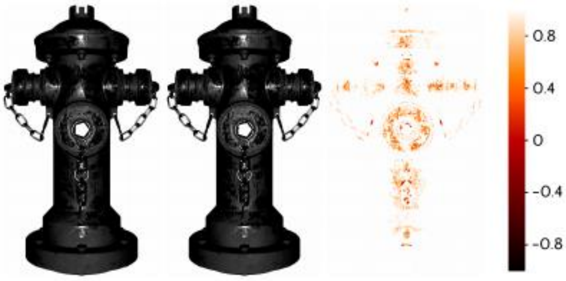

Fig. 4 SVR remapping of SVBRDF from Mitsuba's default Ward implementation to Ward eithin Blender's internal renderer [13].

Practical real-time hex-tiling is a computer graphics technology which is used to generate the effect of hex-tiles in real-time rendering. This technology is usually used in game development to provide better visual effects and higher performance [14]. Barycentric Quad Rasterization is a technique for triangle rendering in computer graphics and it is the most basic geometric primitive, and rasterization is the process of projecting 3D graphics onto a 2D screen [15].

2.3.1 Algorithms and Optimization

Providing an optimization algorithm for Kronecker sequences aimed at generating efficient uniform and diverse multidimensional sampling sequences [16]. Extending 3D SurfaceNets to generate surfaces of segmented 3D medical images composed of multiple materials represented as indexed labels [17]. Simplex Noise is a popular algorithm for generating realistic program textures. Simplex Noise provides smoother and more efficient results than Perlin Noise [18]. The writer designed a fast algorithm to analyze scenes where motion vectors are unavailable and high performance is required. And key element is to effectively embed patch-based correspondence detection into hierarchical algorithm [1]. Progressive lease-squares encoding for linear bases is a coding algorithm for image compression, which can keep high compression rate while preserves the details and quality of the image [19].

Laine-Karra's hash function is used to realize the effect. Owen scrambles Sobol sequence is simple but effective enough to evaluate the sequence with uncertain length and dimension in real-time, also parallel, progressive or adaptive integration is easy to be realized [20]. A new framework for layering and synthesizing bump maps is proposed including supporting multiple groups of texture coordinates and program-generated texture coordinates [21]. An interactive framework is also introduced to design low-poly 3D models from 2D model sheets, which shows how the model (e.g., sketched characters) looks from the front and side [22].

Fig. 5 Quantization impact. All images were rendered using 512 spp[22].

3. Modeling

Modeling refers to the process of creating a digital representation of an object or scene using specialized software tools. In the context of computer graphics, modeling involves defining the shape, appearance, and behavior of virtual objects to be used in simulations, animations, video games, and other visual applications. There are several types of modeling techniques, including Geometric Modeling, Animation Modeling, and so on.

3.1. Animation Models

Motion synthesis and animation models are technologies in computer graphics which can be applied in games, movies, virtual reality, and various other digital media industries. The first framework is proposed to synthesize the whole-body motion of virtual characters, which performs specified actions and places 3D objects in their range [23]. Then a new motion is presented as an adaptive method, allowing real-time generation of the whole-body motion of humanoid avatars in remote space Suggested methods to deal with it [24].

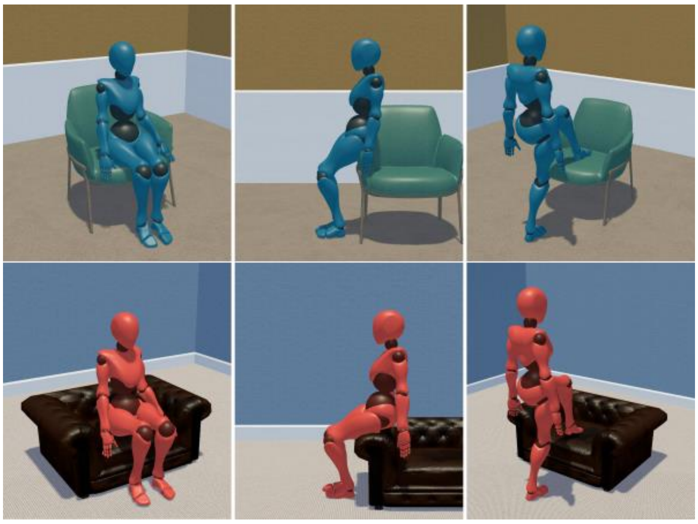

Fig. 6 Top: User motions. Bottom: Adapted avatar motions. [24]

The natural and robust can be learned for manual operation of simple objects in dynamic simulation from high-quality motion capture examples through careful design of imitation learning problems and deep reinforcement learning [25]. SoftSMPL is a learning-based method, which models the real soft tissue dynamics as a function of body shape and motion hard-core, enabling us to simulate highly realistic dynamics [26]. Besides, an interactive 3D face editing method based on depth generation model is proposed. Essentially the model learns to merge components and does not need additional merging process. By optimizing the potential variables according to the vertex position constraints provided by users to realize interactive face modeling [27]. Lastly, the method is proposed to extract the makeup of 3D facial model from a single makeup portrait also the method includes the following three steps which doesn't be showed here [28].

3.2. Models Based on Image

Image processing involves manipulating or enhancing digital images to improve their quality, extract information, or achieve specific visual effects. This can include tasks such as noise reduction, image enhancement, color correction, image segmentation, and object recognition. These techniques play a crucial role in improving image quality, analyzing image content, and enabling advanced image-based applications. A new computing framework is proposed that can automatically generate gold. A gradual method is also put forward, which can effectively solve the optimization problem and produce a series of abstractions to reduce the level of detail to approximate the input. Besides, a new sub-pixel deblurring method is used for low-resolution anti-aliasing grating clip art images and prove that its performance is obviously much more useful than the existing technology [29,30,31].

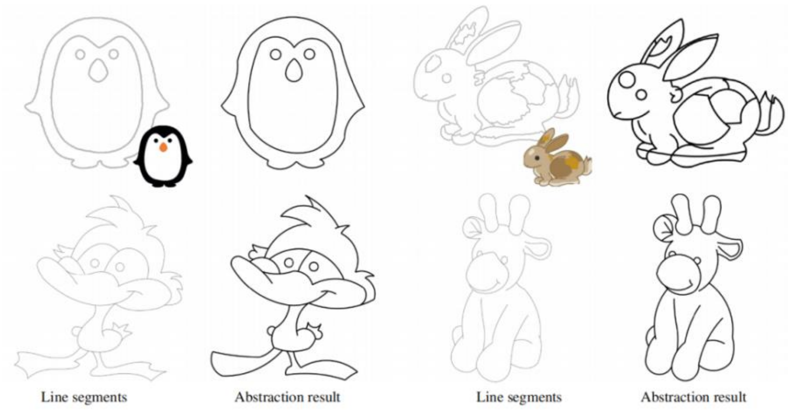

Fig. 7 Visual results.

On the other hand, rendering models are used to generate realistic images from 3D scenes or models. Rendering involves simulating the behavior of light and materials to create visually appealing images that closely resemble real-world objects and environments. Techniques such as ray tracing, rasterization, and global illumination are commonly used in rendering to calculate how light interacts with surfaces, shadows are cast, and reflections are formed. Rendering models are essential for creating realistic visual effects in movies, video games, architectural visualization, and virtual reality applications.

3.3. Geometric Modeling

Geometric modeling is a fundamental concept in computer graphics and computational geometry, involving the creation and manipulation of digital representations of geometric shapes and structures. This field plays an important part in various industries, including computer-aided design (CAD), animation, virtual reality, and computer-aided manufacturing (CAM). Geometric modeling is essential for tasks such as creating 3D models, simulating physical phenomena, and visualizing complex structures. It serves as the foundation for computer-aided design, computer graphics, and various engineering and scientific simulations. As technology advances, geometric modeling continues to evolve, enabling more efficient and realistic representation of complex shapes and structures. Experiments in dynamic non-rigid DFaust and ShapeNet classes show that our problem identification and solution provide the latest development level of unsupervised dense correspondence [32]. In computer graphics and geometry processing, a rotating body is a three-dimensional entity created by the rotation of a curve or contour around an axis. For example, a cylinder, a sphere and so on, which are all rotating bodies [33]. Besides, proposing an extensible and practical algorithm, and demonstrating the results on dense grids are close to those obtained by standard function mapping algorithm in part of the calculation time. In addition, the method is far superior to the standard accelerated procedure [34].

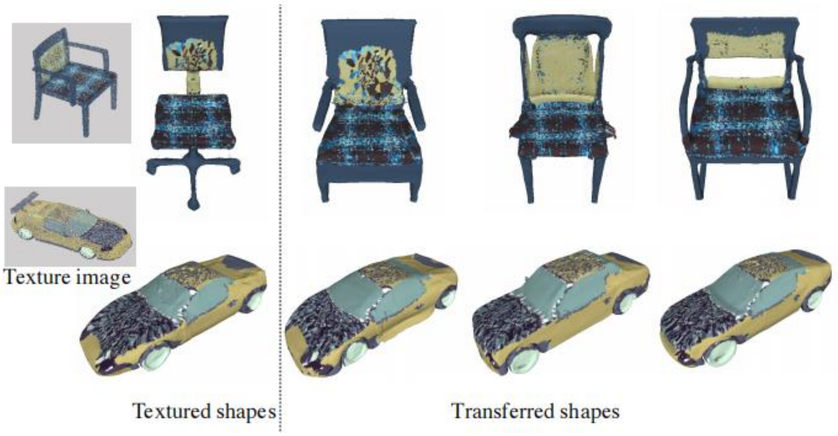

Fig. 8. Texture transfer.

3.4. Physics Modeling

Physics simulation and material modeling are critical components of many scientific and engineering fields. They enable researchers and engineers to understand and predict the behavior of physical systems, materials, and structures under various conditions.

Takeing a pair of two-dimensional shapes as input and optimizing each shape into a gear by locating the center of rotation on each shape, minimally modifying each shape to form the boundary of the gear and constructing teeth suitable for the meshing of the gear. Also, demonstrateing the usability of our approach by generating a rich variety of non-circular gears from various inputs and 3D printing several of them [35]. Material models are crucial for designing and optimizing the performance of components and structures in various industries, including aerospace, automotive, and consumer electronics. Simply transforming the anisotropy problem into an equivalent isotropy problem makes little change to existing simulators, and it allows us to reuse popular anisotropic plasticity models, such as the Drucker-Prager yield criterion, rather than inventing new anisotropic plasticity models for each phenomenon being studied hope simulation [36].

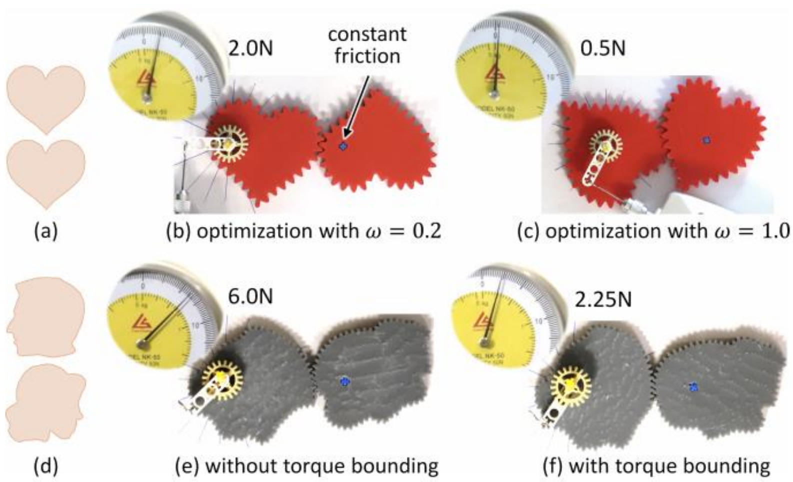

Fig. 9 Experiments on maximum instantaneous torque required to drive the gears.

Overall, physics simulation and material modeling play a pivotal role in advancing scientific knowledge, optimizing engineering designs, and driving innovation across a wide range of industries.

3.5. Machine Learning Models

Data processing and analysis models are tools used to handle and analyze large amounts of data, helping people extract meaningful information and insights from the data. These models include various algorithms and techniques such as statistics, machine learning, artificial intelligence, data mining, and more.

In computer graphics, physics-based optical transport models are a challenge because of the complex light fields. Monte Carlo optical transmission algorithm simulates the transmission process of light in a scene by random sampling. However, the traditional Monte Carlo optical transmission algorithm has problems of efficiency and convergence speed. SMCMLT improves it by introducing hierarchical sampling and Markov chain Monte Carlo method. Specifically, SMCMLT divides the sampling process into multiple stages, each uses a different sampling strategy and weight function [37]. Sparse voxel dag(or dag for short) is investigated to find that it can be used as an alternative method to avoid uv mapping. The result is an algorithm capable of achieving a high compression ratio of voxel structure and color and it can be used for automated systems [38]. Besides, Deep learning techniques are used to generate ambient light maps of the scene, thus reducing computation. NEnv uses neural networks to generate ambient light maps, And it takes a 3D model of the scene as input and outputs a high-resolution ambient light map [39].

(d) COFFEE CART scene.

Fig. 10 Comparision between our method (ROMA), Distance field (DF), and hardware ray tracing (Ref.) on simulating ambient occulusion [40].

4. Discussion

The discussion introduces differant kinds of methods in rendering and modeling, it shows the limitation and advantages, also presents the new direction.

Table 1. the data for different renderingmethods

Rendering	RayTracing Related	GPURay Tracing	Comperision of the method with Halton for the Pool scene with motionblur and find the method in this paper is more useful.	In practice, due to efficiency issues with KD-trees, it is only applicable to low-dimensional spaces (such as 3D to 6D). Another issue is that different dimensions may require different numbers of samples (e.g., image space versus time).
		Rendering signed distance functions (SDFs)	Values are formatted and clamped to [−1, 1] such that a value of 1 encodes 2.5 interior voxel diagonals. This allows the SDF values to be stored as 8-bit snorm integers.	The interpolation normals of our new method use a time ratio of 1-7% Baseline, but from time to time provides a smoother image. Short distance
		hardware ray tracing–based progressive photon mapper (RTPM)	When comparing the render times for the individual render passes, it can be observed that RTPM is the fastest for all of the Photon Generation and Photon Collection	only photons from the other side of the surface can Due to BSDF, this data is shown in black. It's just a If there are more photons on the invisible side of the surface, because
	Real-time Rendering	Monte Carlo rendering.	Breakdown of our timings for the HIMALAYA scene, the sum is 1251	Even in the presence of incoherent motion, usually the result still matches the pixel environment well, but this can lead to some random buzz in the animation.
		GEN-LAL	By observing that GEN-LAL can achieve convergence in about half the iterations required by either the Random or Fully Informed baselines	The GEN-LAL approach has advantages in model performance, but there are some implementation challenges, including the requirements for computing resources and hardware, as well as the need for optimization of loss functions and training processes.
		MMPX	On all platforms MMPX is fast enough to operate within a sprite renderer for scaling and rotation of small numbers of sprites, N/A 0.955 0.156 0.039 0.005	There are some interesting limitations and directions for future work on this issue. For 1:1 gradients,which was able to avoid brightness deviations on thin stripes by corresponding downstream patterns. However, this necessarily creates an inconsistency between light and dark disc shape filtering

Table 2. The data for different kinds of modeling methods

Models with method	Animation Models	IMoS	compare with other motion synthesis methods, the method's AVE is 3.82±0.004, Accuracy 0.87±0.011	The method does not support textures and fine appearance details and cannot be used to produce deceptive content. Then the results are not real designs and cannot be confused with real scenarios.
	Models Based on Image	image abstraction	Comperision of performance using fish example, the time least is 65.37	one limitation is that disconnected line segments cannot be abstracted using one circular arc as done by designers in some special occasions
	Geometric Modeling	Realistic fluid simulation	Scene stactics for 3D results, the mempry is 8.7GB	To investigate new coverage schemes during base building: for example, even if two-power improvement scheme is a natural choice, it may be able to get better coverage from, say, an image's power improvement
	Physics Modeling	MPM	Performance of method:time in ms for each time step of isotropic MPM, and the least is 100k, the most is 200k.	Anisotropic simulators tend to have worse numerical conditions than isotropic simulators because the ratio between eigenvalues is necessarily exaggerated when scaling the material stiffness more along one axis than the other
	Machine learning modeling	Path-space MCMC	The differen metric values. The MAPE of the method is 0.877.	By distributing the samples evenly on the image space, achieves the best score in relative metric, at the cost of introducing more error in the brightest image regions.

5. Conclusion

This article delves into several key aspects of the field of computer graphics, showing significant advances in ray tracing, rendering techniques, algorithm optimization, data processing, and real-time rendering. These advances not only improve the quality and fidelity of computer-generated images, but also open up more possibilities in areas such as virtual reality, augmented reality, and film visual effects. With the convergence of machine learning and deep learning technologies, seeing data-driven approaches have a profound impact on graphics and visualization, providing new ways to simulate complex phenomena, improve rendering, and accelerate graphics processing. Overall, this paper demonstrates the diversity and dynamism of the current field of computer graphics. In the future, as technology continues to evolve, we can expect to see more innovations and breakthroughs in this area.

References

[1] Johannes Hanika, Lorenzo Tessari, and Carsten Dachsbacher, Fast temporal reprojection without motion vectors, Journal of Computer Graphics Techniques (JCGT), vol. 10, no. 3, 19-45, 2021 Available online http://jcgt.org/published/0010/03/02/

[2] Kim, SangBin (Department of Computer Science and Engineering, Korea University, Seoul, Korea, Republic of); Park, Inbum; Kwon, Seongsu;Motion Retargetting based on Dilated Convolutions and Skeleton-specific Loss Functions Han, JungHyun Source: Computer Graphics Forum, v 39, n 2, p 497-507, May 1, 2020 https://doi.org/10.1111/cgf.13947

[3] Viktor Enfeldt and Prashant Goswami, A Polarizing Filter Function for Real-Time Rendering, Journal of Computer Graphics Techniques (JCGT), vol. 10, no. 2, 59-82, 2021. Available online http://jcgt.org/published/0010/02/03/

[4] Mark Bo Jensen, Jeppe Revall Frisvad, and J. Andreas Bærentzen, Performance Comparison of Meshlet Generation Strategies, Journal of Computer Graphics Techniques (JCGT), vol. 12, no. 2, 1-27, 2023 Available online http://jcgt.org/published/0012/02/01/.

[5] Daniel Meister and Toshiya Hachisuka, Lightweight Multidimensional Adaptive Sampling for GPU Ray Tracing, Journal of Computer Graphics Techniques (JCGT), vol. 11, no. 3, 43-64, 2022. Available online http://jcgt.org/published/0011/03/03/

[6] René Kern, Felix Brüll, and Thorsten Grosch, Accelerated Photon Mapping for Hardware-based Ray Tracing, Journal of Computer Graphics Techniques (JCGT), vol. 12, no. 1, 1-28, 2023. Available online http://jcgt.org/published/0012/01/01/

[7] Herman Hansson-Söderlund, Alex Evans, and Tomas Akenine-Möller, Ray Tracing of Signed Distance Function Grids, Journal of Computer Graphics Techniques (JCGT), vol. 11, no. 3, 94-113, 2022. Available online http://jcgt.org/published/0011/03/06/

[8] Manuel N. Gamito, Ray Traversal of OpenVDB Frustum Grids, Journal of Computer Graphics Techniques (JCGT), vol. 10, no. 1, 49-63, 2021. Available online http://jcgt.org/published/0010/01/03/

[9] Norbert Bus and Tamy Boubekeur, Double Hierarchies for Directional Importance Sampling in Monte Carlo Rendering, Journal of Computer Graphics Techniques (JCGT), vol. 6, no. 3, 25-37, 2017Available online http://jcgt.org/published/0006/03/02/

[10] Takahiro Kuge, Tatsuya Yatagawa, and Shigeo Morishima, Real-time Shading with Free-form Planar Area Lights using Linearly Transformed Cosines, Journal of Computer Graphics Techniques (JCGT), vol. 11, no. 1, 1-16, 2022. Available online http://jcgt.org/published/0011/01/01/

[11] Carmelo J. Fdez-Agüera, A Multiple-Scattering Microfacet Model for Real-Time Image Based Lighting, Journal of Computer Graphics Techniques (JCGT), vol. 8, no. 1, 45-55, 2019. Available online http://jcgt.org/published/0008/01/03/

[12] Alisa Jung, Johannes Hanika, and Carsten Dachsbacher, Detecting Bias in Monte Carlo Renderers using Welch’s t-test, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 2, 1-25, 2020Available online http://jcgt.org/published/0009/02/01/

[13] Alejandro Sztrajman, Jaroslav Křivánek, Alexander Wilkie, and Tim Weyrich, Image-based Remapping of Spatially varying Material Appearance, Journal of Computer Graphics Techniques (JCGT), vol. 8, no. 4, 1-30, 2019Available online http://jcgt.org/published/0008/04/01/

[14] Morten S. Mikkelsen, Practical Real-Time Hex-Tiling, Journal of Computer Graphics Techniques (JCGT), vol. 11, no. 3, 77-94, 2022 Available online http://jcgt.org/published/0011/03/05/

[15] Gregor Mückl and Carsten Dachsbacher, Real-time Room Reverb in Large Scenes using Transport Path Precomputation, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 1, 33-53, 2020 Available online http://jcgt.org/published/0009/01/03/.

[16] Thomas Roughton, Progressive Least-Squares Encoding for Linear Bases, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 1, 17-32, 2020 Available online http://jcgt.org/published/0009/01/02/

[17] David Vega, Javier Abache, and David Coll, A Fast and Memory Saving Marching Cubes 33 Implementation with the Correct Interior Test, Journal of Computer Graphics Techniques (JCGT), vol. 8, no. 3, 1-18, 2019Available online http://jcgt.org/published/0008/03/01/

[18] Théo Jonchier, Alexandre Derouet-Jourdan, and Marc Salvati, Implementation of Fast and Adaptive Procedural Cellular Noise, Journal of Computer Graphics Techniques (JCGT), vol. 8, no. 1, 35-44, 2019 Available online http://jcgt.org/published/0008/01/02/

[19] Mads J. L. Rønnow, Ulf Assarsson, Erik Sintorn, and Marco Fratarcangeli, Improved Accuracy for Prism-Based Motion Blur, Journal of Computer Graphics Techniques (JCGT), vol. 11, no. 1, 82-95, 2022 Available online http://jcgt.org/published/0011/01/05/

[20] Sylvain Rousseau and Tamy Boubekeur, Unorganized Unit Vectors Sets Quantization, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 4, 21-37, 2020. Available online http://jcgt.org/published/0009/04/02/

[21] Brent Burley, Practical Hash-based Owen Scrambling, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 4, 1-20, 2020 Available online http://jcgt.org/published/0009/04/01/Morten S. Mikkelsen, Surface Gradient–Based Bump Mapping Framework, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 3, 60-90, 2020. Available online http://jcgt.org/published/0009/03/04/

[22] Tsukasa Fukusato, Seung-Tak Noh, Takeo Igarashi, and Daichi Ito, Interactive Meshing of User-Defined Point Sets, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 3, 39-58, 2020 Available online http://jcgt.org/published/0009/03/03/

[23] Joel Castellon, Moritz Bächer, Matt McCrory, Alfredo Ayala, Jeremy Stolarz, and Kenny Mitchell, Active Learning for Interactive Audio-Animatronic Performance Design, Journal of Computer Graphics Techniques (JCGT), vol. 9, no. 3, 1-19, 2020 Available online http://jcgt.org/published/0009/03/01/

[24] Ghosh, Anindita (German Research Center for Artificial Intelligence (DFKI), Germany); Dabral, Rishabh; Golyanik, Vladislav; Theobalt, Christian; IMoS: Intent-Driven Full-Body Motion Synthesis for Human-Object Interactions Slusallek, Philipp Source: Computer Graphics Forum, v 42, n 2, p 1-12, May 2023

[25] Choi, Soojin (Visual Media Lab, KAIST, Korea, Republic of); Hong, Seokpyo; Cho, Kyungmin; Kim, Chaelin; Online Avatar Motion Adaptation to Morphologically-similar Spaces Noh, Junyong Source: Computer Graphics Forum, v 42, n 2, p 13-24, May 2023 https://doi.org/10.1111/cgf.14740

[26] Zhang, Yunbo (Georgia Institute of Technology, United States); Clegg, Alexander; Ha, Sehoon; Turk, Greg; Learning to Transfer In-Hand Manipulations Using a Greedy Shape Curriculum Ye, Yuting Source: Computer Graphics Forum, v 42, n 2, p 25-36, May 2023 https://doi.org/10.1111/cgf.14741

[27] Santesteban, Igor (Universidad Rey Juan Carlos, Madrid, Spain); Garces, Elena; Otaduy, Miguel A.; SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Casas, Dan Source: Computer Graphics Forum, v 39, n 2, p 65-75, May 1, 2020, https://doi.org/10.1111/cgf.13912

[28] Face Editing Using Part-Based Optimization of the Latent Space. Aliari, Mohammad Amin (Ecole de technologie supérieure, Montreal, Canada); Beauchamp, Andre; Popa, Tiberiu; What's in a Decade? Transforming Faces Through TimePaquette, Eric Source: Computer Graphics Forum, v 42, n 2, p 269-279, May 2023 https://doi.org/10.1111/cgf.14760

[29] Yang, Xingchao (CyberAgent, AI Lab, Japan); Taketomi, Takafumi; Makeup Extraction of 3D Representation via Illumination-Aware Image Decomposition Kanamori, Yoshihiro Source: Computer Graphics Forum, v 42, n 2, p 293-307, May 2023. https://doi.org/10.1111/cgf.14762

[30] Hsiao, Kai-Wen (National Tsing HuaUniversity, Taiwan); Yang, Yong-Liang; Chiu, Yung-Chih; Hu, Min-Chun; Yao, Chih-Yuan; Img2Logo: Generating Golden Ratio Logos from ImagesChu, Hung-Kuo Source: Computer Graphics Forum, v 42, n 2, p 37-49, May 2023 https://doi.org/10.1111/cgf.14742

[31] Matusovic, Marko (Delft University of Technology, Netherlands); Parakkat, Amal Dev; Interactive Depixelization of Pixel Art through Spring Simulation Eisemann, Elmar Source: Computer Graphics Forum, v 42, n 2, p 51-60, May 2023 https://doi.org/10.1111/cgf.14742

[32] Subpixel Deblurring of Anti-Aliased Raster Clip-Art Yang, J. (University of British Columbia, Canada); Vining, N.; Kheradmand, S.; Carr, N.; Sigal, L.; Sheffer, A. Source: Computer Graphics Forum, v 42, n 2, p 61-76, May 2023 https://doi.org/10.1111/cgf.14744

[33] Parallel Transformation of Bounding Volume Hierarchies into Oriented Bounding Box Trees Vitsas, N. (Department of Informatics, Athens University of Economics and Business, Greece); Evangelou, I.; Papaioannou, G.; Gkaravelis, A. Source: Computer Graphics Forum, v 42, n 2, p 245-254, May 2023 https://doi.org/10.1111/cgf.14758

[34] Unsupervised Template Warp Consistency for Implicit Surface Correspondences. Liu, Mengya (ETH Zurich, Switzerland); Chhatkuli, Ajad; Postels, Janis; Gool, Luc Van; Tombari, Federico Source: Computer Graphics Forum, v 42, n 2, p 77-87, May 2023. https://doi.org/10.1111/cgf.14745

[35] [35]Son, S.-H. (Dept. of Computer Science and Eng., Seoul National University, Korea, Republic of); Yoon, S.-H.; Kim, M.-S.;Efficient Minimum Distance Computation for Solids of Revolution Elber, G. Source: Computer Graphics Forum, v 39, n 2, p 535-544, May 1, 2020 https://doi.org/10.1111/cgf.13950

[36] Scalable and Efficient Functional Map Computations on Dense Meshes. Magnet, Robin (LIX, Ecole Polytechnique, IP Paris, France); Ovsjanikov, Maks Source: Computer Graphics Forum, v 42, n 2, p 89-101, May 2023https://doi.org/10.1111/cgf.14746

[37] Xu, Hao (The Chinese University of Hong Kong, Hong Kong); Fu, Tianwen; Song, Peng; Zhou, Mingjun; Fu, Chi-Wing; Computational Design and Optimization of Non-Circular Gears Mitra, Niloy J. Source: Computer Graphics Forum, v 39, n 2, p 399-409, May 1, 2020 https://doi.org/10.1111/cgf.13939

[38] Gruson, Adrien (The University of Tokyo, Japan); West, Rex; 1. Stratified Markov Chain Monte Carlo Light Transport Hachisuka, Toshiya Source: Computer Graphics Forum, v 39, n 2, p 351-362, May 1, 2020, https://doi.org/10.1111/cgf.13935

[39] Dolonius, D. (Chalmers University of Technology, Sweden); Sintorn, E.; UV-free Texturing using Sparse Voxel DAGs Assarsson, U. Source: Computer Graphics Forum, v 39, n 2, p 121-132, May 1, 2020 https://doi.org/10.1111/cgf.13917

[40] Rodriguez-Pardo, Carlos (SEDDI, Madrid, Spain); Fabre, Javier; Garces, Elena; NEnv: Neural Environment Maps for Global IlluminationLopez-Moreno, Jorge Source: Eurographics Symposium on Rendering, v 42, n 4, 2023 https://doi.org/10.1111/cgf.14883