Seminar topics

Advanced Computer Graphics, 2024/25

1 Introduction

The document describes the seminar topics for the Advanced Computer Graphics course given in 2024/25. The seminar work must be completed during the spring semester and defended prior to the final deadline. Students must defend the seminar at the laboratory exercises and present it at the final presentation session in front of the class.

Upon successful completion, the seminar may contribute up to 30 % to the final grade. It must be graded with more than 15 % for a positive grade. The defense of the seminar after the deadline lowers its maximum contribution: 1-week extension: 22.5 %, 2-week extension 17.5 %. Seminars can also be completed and defended prior to the given deadline.

When the topics are assigned, the students must report their GitHub username and initialize the repository. During the semester the students must present their progress every two weeks through git commits at the laboratory exercises.

The expected outputs of each seminar are the seminar report and code repository published on GitHub under the CC-BY, BSD or MIT license. The report must be submitted through the FRI Učilnica. It must be written as a research paper in accordance with the Eurographics LaTeX template.

2 Seminar Topics

2.1 Hybrid volume and surface rendering

Photorealism in rendering heavily depends on how materials are defined, sampled, and applied within a scene. A scene typically consists of surfaces and volumes, which must be handled separately by the rendering software due to their largely incompatible mathematical definitions. While surface appearance is governed by the BSDF, the appearance of volumes is determined by the phase function. Kroes et al. [1] demonstrated that these two seemingly distinct approaches can be seamlessly combined by locally estimating the degree to which a region resembles a surface or a volume and then merging their scattering distributions using a weighted sum or applying multiple importance sampling to their respective sampling strategies. In this seminar, you will explore several open aspects of hybrid volume and surface rendering, including the impact of different multiple importance sampling strategies on convergence, different techniques for estimating surface normals, and alternative methods for assigning weights to the two scattering distributions.

References:

[1] T. Kroes, F. H. Post, C. P. Botha, "*Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework,*" PloS one, vol. 7, no. 7, 2012, doi: 10.1371/journal.pone.0038586.

2.2 Radiance probes for global illumination in volume rendering

Accurate and efficient computation of global illumination is a significant challenge in volume rendering. The appearance of materials is heavily influenced by the surrounding environment, making the quality of rendering depend on effective environment sampling. While path tracing can produce accurate results, it is often too slow for real-time execution. A common practical solution is to precompute radiance at fixed points in space known as radiance probes or light probes and use these values during rendering to avoid costly computation for every light ray. In this seminar, you will explore the radiance probe method and implement it in a prototype application. You will evaluate the results both qualitatively and quantitatively and compare them to path tracing.

References:

[1] R. Khlebnikov, P. Voglreiter, M. Steinberger, B. Kainz, D. Schmalstieg, "*Parallel Irradiance Caching for Interactive Monte-Carlo Direct Volume Rendering,*" Computer Graphics Forum, vol. 33, no. 3, 2014, doi: 10.1111/cgf.12362.

2.3 Edgebreaker for arbitrary meshes

Polygonal models often contain a significant amount of redundant information, particularly in encoding connectivity data. This seminar focuses on compressing triangular meshes using the Edgebreaker algorithm. Edgebreaker traverses the mesh in a way that each new triangle is typically next to an already encoded one, allowing efficient differential encoding of vertex attributes. However, since Edgebreaker is limited to compressing manifold meshes, the implementation should be extended to support arbitrary meshes such as disjoint, non-manifold, and non-orientable meshes.

References:

[1] J. Rossignac, "*Edgebreaker: connectivity compression for triangle meshes,*" in IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 1, pp. 47-61, 1999, doi: 10.1109/2945.764870.

2.4 Volumetric special effects in games

In modern games, special effects like explosions are typically simulated using particle systems and rendered as sprites or simple textured meshes. While easy to implement, these methods often produce noticeable visual artifacts where transparent sprites intersect with non-transparent geometry. This seminar explores modeling special effects using volumetric simulation and rendering. The volumes should be low-resolution enough to enable real-time simulation while being upscaled with procedural details during rendering. Additionally, proper rendering of shadows and self-shadowing is essential for achieving realistic results.

References:

[1] https://www.youtube.com/watch?v=ryB8hT5TMSg
[2] https://www.scratchapixel.com/lessons/3d-basic-rendering/volume-rendering-for-developers/intro-volume-rendering.html
[3] https://bronsonzgeb.com/index.php/2021/05/29/gpu-mesh-voxelizer-part-2/
[4] https://advances.realtimerendering.com/s2015/The%20Real-time%20Volumetric%20Cloudscapes%20of%20Horizon%20-%20Zero%20Dawn%20-%20ARTR.pdf
[5] https://shaderbits.com/blog/creating-volumetric-ray-marcher
[6] https://iquilezles.org/articles/distfunctions/
[7] https://github.com/GarrettGunnell/CS2-Smoke-Grenades

2.5 Hierarchical clustering for point cloud compression

Point cloud datasets are growing larger due to the increasing accessibility of acquisition technology. As a result, many countries are investing in the periodic acquisition of airborne LiDAR data. This data serves various purposes, including land use monitoring in agriculture, forest management, urban planning, and landslide tracking. However, such datasets are typically massive (e.g., several terabytes for the area of Slovenia [1]), making compression a crucial factor. While multiple compression techniques exist, there is still room for improvement, particularly because the choice of method depends on the specific use case. In this seminar, we aim to explore how hierarchical clustering [2], based on k-NN [3], can be leveraged for compressing these datasets. Our focus is on a visualization use case, where we seek to enable efficient rendering of country-wide datasets with level-of-detail support during loading while preserving as many visual aspects as possible. This seminar will be co-mentored by Yannick Kuhar from the Laboratory of Algorithmics.

References:

[1] https://gis.arso.gov.si/evode/
[2] Bansal, Nikhil, Vincent Cohen-Addad, Milind Prabhu, David Saulpic, and Chris Schwiegelshohn. "Sensitivity Sampling for $ k $-Means: Worst Case and Stability Optimal Coreset Bounds." *arXiv preprint arXiv:2405.01339* (2024).
[3] Arthur, David, and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Stanford, 2006.

2.6 Ray tracing Gaussians

Gaussian splatting is a popular and versatile approach to scene representation and rendering. The scene is filled with particles which have a position, rotation, scale, and color, and their influence is determined by a Gaussian function. Such a simplistic representation can be taken advantage of during rendering by transforming each particle into a view-aligned billboard and using the rasterization pipeline. However, the particles must be ordered according to depth to ensure correct transparency calculations, which is computationally expensive and introduces popping artifacts when the order suddenly changes due to camera rotation. Moenne-Loccoz et al. [1] introduce a ray tracing algorithm for rendering Gaussians by calculating a closed-form solution for transmittance. This allows the use of complex camera models and relighting through recursive ray tracing. Your task is to implement the volumetric rendering algorithm presented in their paper.

References:

[1] J. Condor, S. Speierer, L. Bode, A. Božič, S. Green, P. Didyk, A. Jarabo, "*3 Don't Splat your Gaussians: Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media,*" ACM Transactions on Graphics, vol. 44, no. 1, 2025, doi: 10.1145/3711853.

2.7 3D Gaussian Splatting upgrades

Gaussian splatting [1] has become a very popular and widely used approach for visualizing 3D scenes. While the optimization stage for preparation of Gaussian scene representation is non-trivial, the rendering stage – splatting – is fairly simple. Therefore, the idea of this seminar is to improve the splatting technique used for rendering. One possible way is to use different kernels as shown by Chen et al. in [2]. Another is to use a mip-like approach for alias-free results as presented by Yu et al. [3, 4]. One more improvement can be the deblurring as presented by Lee et al. [5]. Select one of the mentioned methods or propose your own improvement and implement it.

References:

[1] Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "*3D gaussian splatting for real-time radiance field rendering*." ACM Trans. Graph. 42, no. 4 (2023): 139-1.

[2] Chen, Haodong, Runnan Chen, Qiang Qu, Zhaoqing Wang, Tongliang Liu, Xiaoming Chen, and Yuk Ying Chung. "*Beyond Gaussians: Fast and High-Fidelity 3D Splatting with Linear Kernels*." arXiv preprint arXiv:2411.12440 (2024).

[3] Yu, Zehao, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. "*Mip-splatting: Alias-free 3d gaussian splatting*." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19447-19456. 2024.

[4] https://github.com/autonomousvision/mip-splatting

[5] Lee, Byeonghyeon, Howoong Lee, Xiangyu Sun, Usman Ali, and Eunbyung Park. "*Deblurring 3d gaussian splatting*." In European Conference on Computer Vision, pp. 127-143. Cham: Springer Nature Switzerland, 2024.

2.8 Tree log remeshing from Gaussian representation

In this seminar, you will develop a method for generating detailed 3D meshes of tree logs using a Gaussian representation. Given a large dataset of high-resolution tree log photos, the first step will be to train a Gaussian model to capture the underlying structure and surface details. The trained representation will then be converted into a mesh with adjustable precision, allowing for optimization based on different use cases and computational constraints. The focus of this seminar will be on refining the remeshing process to balance detail and efficiency while preserving realistic surface characteristics.

2.9 Procedural generation of tree log models

In this seminar, you will develop a program for procedurally generating 3D models of tree logs based on user-defined parameters. The program should take inputs such as log length, diameter, eccentricity, surface roughness, curvature, tapering, number and shape of gnarls, and other relevant characteristics (a full list of parameters can be obtained from the teaching assistant). Using these parameters, the program should generate a detailed 3D model of a tree log, including a realistic texture that reflects the natural variations found in real wood. The goal is to create a flexible system capable of producing diverse and lifelike log models suitable for use in rendering, simulations, or game environments.

2.10 Access road tracing and analysis

In this seminar, you will design an application for tracing and analyzing access roads on uneven terrain. Logging operations often require on-site planning of new access roads for transporting tree logs. It is essential to explore multiple routing possibilities and analyze key road characteristics such as length, inclination, profile, and estimated construction requirements. The application should display an orthophoto of the terrain along with the user's position, obtained via GPS. Users should be able to trace road trajectories and visualize their characteristics. Elevation data can be sourced from either a digital elevation model or a LIDAR scan. Ideally, the application will be web-based and accessible through a modern browser.

2.11 Special topic (with prior agreement)

If you have a specific topic in mind that aligns with the course syllabus but is not covered by the topics above, you may propose your idea to the professor or teaching assistant. Before the topic selection deadline, you must discuss your proposal with them and obtain their approval for it to be considered as a seminar topic.