Seminar topics

Advanced Computer Graphics, 2019/20

1 Introduction

The document describes the seminar topics for the Advanced Computer Graphics course given in 2019/20. 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.

Students have to present their GitHub username and initialize the repository for seminar code. During the semester students have to 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 Ucilnica. It must be written as a research paper in accordance with the Eurographics LaTeX template[1].

2 Seminar Topics

2.1 Cloth simulation for wearing scenarios

Finding out which clothing suits the selected individual is a personalized process that involves trying out several pieces one after another or getting personally tailored clothes after specific measurement procedure. We want to test whether it is possible to estimate how well a piece of clothing fits an individual through simulation process. To do this you will have to set-up a simulation environment (e.g. ARCSim[2] [1]), define/model a specific clothing (e.g. T-Shirt, Long sleeved shirt, etc.), obtain a virtual human body model, define the initial setups and run multiple simulations. To evaluate the fitting we want to estimate the stretching of the fabric on the body. This can be done by comparing triangulation on relaxed clothing and fitted one. The stretching should be visualized as a heatmap texture on the clothing model. As part of the seminar you need to develop a graphical tool for setting up the simulation scenario as well as for visualization of the simulation results together with heatmaps.

References:

[1] R. Narain, A. Samii, and J. F. O'Brien, "*Adaptive Anisotropic Remeshing for Cloth Simulation*", ACM Transactions on Graphics, 31(6):147:1–10, 2012.

[2] H. Wang, R. Ramamoorthi, and J. F. O'Brien, "*Data-Driven Elastic Models for Cloth: Modeling and Measurement*", 30(4):71:1–11, 2011.

2.2 Procedural generation of volumetric fluid environments

The goal of the seminar is to create a procedural model of fluid environment which can be used in volumetric rendering environment VPT [1]. The basic volumetric environment should be generated using 3D Perlin noise where noise values present pressure levels in volume. Furthermore, extend the procedural fluid model by applying simple fluid simulation based on starting conditions defined by initial Perlin noise [2, 3]. For even better final results try to implement parts of the work presented in [4]. The final goal is to generate a static volume containing fluid environment with non-transparent ground material and wavy surface.

References:

[1] https://www.overleaf.com/gallery/tagged/eurographics

[2] http://graphics.berkeley.edu/resources/ARCSim/

[1] Ž. Lesar, C. Bohak, and M. Marolt, "*Real-time interactive platform-agnostic volumetric path tracing in WebGL 2.0*", Web3D 2018: proceedings, 2018.
[2] J. Stam, "*Stable Fluids*", ACM Transactions on Graphics, 1999.
[3] J. Stam, "*Real-Time Fluid Dynamics for Games",* Game Developers Conference, 2003.
[4] J. G. Magnus; S. Bruckner, "*Interactive Dynamic Volume Illumination with Refraction and Caustics*", IEEE Transactions on Visualization and Computer Graphics, 2018.

2.3 Procedural cell shape model

The goal of this seminar is to create a procedural model of call membrane shape extracted from real-life cell data. The cell data can be obtained from Allan Cell institute database [1], where thousands of cells are labeled and their shape can be extracted and used for defining the membrane model parameters. Use statistical shape modeling (e.g. [3]) to define a cell shape. The model must allow users to customize the basic cell shape parameters while retaining the randomness of shape through model seed parameter.

References:

[1] Membrane (CAAX). url: https://www.allencell.org/data-downloading.html
[2] A. Kessel, "*The Living Cell Gallery*". url: https://amit1b.wordpress.com/the-molecules-of-life/10-the-living-cell-gallery/
[3] T. Vrtovec, D. Tomaževič, B. Likar, L. Travnik, F. Pernuš, "*Automated Construction Of 3D Statistical Shape Models*", Image Analysis & Stereology, 2004.

2.4 Transfer function design tool

Creating and adjusting a transfer function for good visualization in volume rendering can be quite demanding. To ease the process and bring the transfer function closer to the everyday user, your task is to develop an exploratory tool which will allow users to easily create and adjust transfer function for specific data. The tool should support display of 2D histogram of a 2D transfer function, where user could easily identify most pronounced values and gradient values in the data. Next, a tool should allow automatic identification of highest peaks in the 2D histogram and allow user to define color and alpha values at those peaks. Once the basic transfer function is defined the tool should allow "browsing" through variations of the transfer function where different parameters would be trimmed (i.e. contrast, blur, levels, etc.).

References:

[1] C. R. Salama, M. Keller, P. Kohlmann, "*High-Level User Interfaces for Transfer Function Design with Semantics*", IEEE Transactions on Visualization and Computer Graphics 2006, doi: 10.1109/TVCG.2006.148.
[2] R. Maciejewski, I. Woo, W. Chen, D. Ebert, "Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation", IEEE Transactions on Visualization and Computer Graphics 2009, doi: 10.1109/TVCG.2009.185.
[3] Y. Liu, C. Lisle and J. Collins, "*Quick2Insight: A user-friendly framework for interactive rendering of biological image volumes,*" 2011 IEEE Symposium on Biological Data Visualization (BioVis)., 2011, doi: 10.1109/BioVis.2011.6094041.

2.5 Volumetric ray casting implementation in VPT

The goal of this seminar is to implement simple volumetric ray casting in VPT [1] framework. The VPT framework is volumetric rendering framework for interactive visualization of volumetric data in web browser. There are several techniques that are already implemented in the framework, however the task of this seminar is to add volumetric ray casting with transfer function support. More about ray casting can be found in [2, 3].

References:

[1] Ž. Lesar, C. Bohak, and M. Marolt, *"Real-time interactive platform-agnostic volumetric path tracing in WebGL 2.0",* Web3D 2018: proceedings, 2018.
[2] Wikipedia: *"Volume ray casting"*, url: https://en.wikipedia.org/wiki/Volume_ray_casting
[3] J. Pawasauskas, *"Volume Visualization With Ray Casting"*, url: http://web.cs.wpi.edu/~matt/courses/cs563/talks/powwie/p1/ray-cast.htm

2.6 Comparison of transmittance and distance estimators in light transport

There are numerous approaches to sample distances and estimating transmittance in heterogeneous media. In 2013, Galtier et al. [1] published a general integral formulation of the light transport problem, postulating an infinite family of estimators. Some of them have been studied and understood, each having their own pros and cons. The goal of this seminar is to study, implement and compare different estimators, preferably in an existing volume rendering framework such as VPT.

References:

[1] M. Galtier, S. Blanco, C. Caliot, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, J. Gautrais, A. Khuong, B. Piaud, and G. Terrée, "*Integral formulation of null-collision Monte Carlo algorithms*", Journal of Quantitative Spectroscopy and Radiative Transfer, 2013, doi: 10.1016/j.jqsrt.2013.04.001

[2] J. Novák, I. Georgiev, J. Hanika, J. Křivánek, and W. Jarosz, "*Monte Carlo methods for physically-based volume rendering*", ACM SIGGRAPH 2018 Courses, ACM Press, 2018, doi: 10.1145/3214834.3214880

[3] J. Novák, I. Georgiev, J. Hanika, J. Křivánek, and W. Jarosz, "*Monte Carlo methods for volumetric light transport simulation*", Computer Graphics Forum, 2018, doi: 10.1111/cgf.13383

2.7 Removing shadows from ortho-photo images

The goal of this seminar is to implement algorithm that will take as an input aerial ortho-photo image and output image with removed shading/shadows. To do this you may apply one of the algorithm presented in [1, 2] or some other state-of-the art shadow removing algorithm. The developed algorithm should be able to read images in different formats (at least JPG and PNG) and output the image in same format. The application shoulf support piping so it can be used in some image processing pipeline.

References:

[1] S. H. Khan, M. Bennamoun, F. Sohel and R. Togneri, *"Automatic Shadow Detection and Removal from a Single Image",* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, doi: 10.1109/TPAMI.2015.2462355.

[2] G. D. Finlayson, S. D. Hordley, M. S. Drew, *"Removing Shadows from Images"*, Proceedings of the 7th European Conference on Computer Vision, 2002, url: https://dl.acm.org/doi/10.5555/645318.649239

2.8 Scattered data approximation with multigrid relaxation

In many cases in computer graphics, we are faced with unordered samples, such as point clouds or path tracing rays. Given a finite set of samples, our task is to guess the function values in the entire domain. Many methods exist for this task, taking significantly different approaches. One of them involves solving Laplacian equation on a grid (e.g. pixels in an image). However, just straightforwardly discretizing the equation results in slow convergence. The goal of this seminar is to explore and implement a multigrid approach (preferably on a GPU), where slow convergence is avoided by discretizing the equation in different resolutions and solving them systematically.

References:

[1] K. Anjyo, J. P. Lewis, and F. Pighin, "*Scattered data interpolation for computer graphics*", ACM SIGGRAPH 2014 Courses, ACM Press, 2014, doi: 10.1145/2614028.2615425

2.9 MERL BRDF support for Path Tracing Framework

The goal of this seminar is to add support for MERL BRDF material definitions into the Path Tracing Framework that you will use during the course. The materials can be found in MERL BRDF Database [1], where one can also find the C++ implementation or material reader.

References:

[1] W. Matusik, H. Pfister, M. Brand, L. McMillan, "*A Data-Driven Reflectance Model*", ACM Transactions on Graphics, 2003, url: https://www.merl.com/brdf/

2.10 Adding support for temporal volume rendering in VPT

The goal of this seminar is to extend the functionality of VPT [1] with support for rendering temporal volumetric data. The added temporal support must support two use cases: (1) fixed-time rendering of individual frame, where system spends equal rendering time for each frame and stores the final render in image sequence, and (2) fixed error rate rendering of individual frame, where each frame is rendered for as long as the difference between the consecutive rendering epochs is below the defined error threshold. As part of this seminar also create 4D temporal volumetric data e.g. smoke simulation or fluid simulation which or use appropriate 4D radiological data e.g. [2].

References:

[1] B. She, P. Boulanger, M. L. Noga, "*Real-Time Rendering of Temporal Volumetric Data on a GPU*", International Conference on Information Visualisation, 2011.
[2] 4D Lungs dataset. url: https://wiki.cancerimagingarchive.net/display/Public/4D-Lung

2.11 Reconstruction of bridges in aerial LiDAR point cloud data

Capturing the world with LiDAR technology gives us an opportunity to get a more detailed insight into the landscape, without the need to walk through vast open spaces or over impassable areas. Since the data are captured from airplanes, we are limited to the representation of upper landscape layers that do not pass through the laser beams. This is the reason that sometimes we do not get a good representation of some parts of the landscape, e.g. bridges, overhangs, building walls or the details underneath the canopy. The goal of this seminar is to reconstruct and repair bridge representations in the point cloud data. The idea is to add new points to the point cloud which will better define the look of the individual bridge, e.g. points on the river surface under the bridge, points on bridge geometry etc. You can use other available geodetic data for obtaining additional information for better reconstruction at Geoportal[3] or Portal Prostor[4].

References:

[1] B. Wu, B. Yu, Q. Wu, S. Yao, F. Zhao, W. Mao, J. Wu *"A Graph-Based Approach for 3D Building Model Reconstruction from Airborne LiDAR Point Clouds"*, 2017. https://www.mdpi.com/2072-4292/9/1/92/htm
[2] M. He, Y. Cheng, Y. Nie, L. Qiu, Z. Zhao *"An Algorithm of Reconstructing LiDAR Building Models Based on Isoheight"*, 2017.
[3] M. Kadaa, L. McKinley *"3D Building Reconstruction from Lidar Based on a Cell Decomposition Approach"*, 2017.
[4] R. Cao, Y. Zhang, X. Liu, Z. Zhao *"3D building roof reconstruction from airborne LiDAR point clouds: a framework based on a spatial database"*, 2017.

2.12 Real-time natural neighbor interpolation of point cloud data using GPU

Interpolation is a way of reconstructing unknown values inbetween known samples. We can use it to reconstruct or compress images and volumes. A good method for interpolating scattered data (samples don't form a regular grid) is natural neighbor interpolation, which uses an underlying voronoi tessellation to keep the interpolation smooth and local. The goal of the seminar is to write an algorithm that applies the method quickly and incrementally (update the voronoi structure when additional samples are added) on a regular grid of target samples. The interpolation would be performed for each target sample in parallel using the GPU via WebGL 2.0 API.

References:

[1] R. Sibson, *"A brief description of natural neighbor interpolation (Chapter 2)"*, 1981.
[2] S. W. Park, L. Linsen, O. Kreylos, J. D. Owens and B. Hamann, "*Discrete Sibson interpolation"*, 2006.
[3] A. Tsidaev, "*Parallel algorithm for natural neighbor interpolation"*, 2016.
[4] J. Živković, "*Natural neighbor interpolation of images and volumes"*, Seminar report, 2019.
[5] A. Košir, *"Natural Neighbor interpolation"*, Seminar report, 2019.

[3] http://www.geoportal.gov.si/slo/
[4] https://www.e-prostor.gov.si/

2.13 Special topic [with prior agreement]

If you have in mind a specific topic which is in line with the course syllabus and is not covered in any of above topics you may present your idea to the Professor or Teaching Assistant and discuss with them whether your idea can be defined as a seminar topic for this course. You must contact and discuss your topic prior to the deadline of topic selection and you have to get the agreement for the topic from the Professor or Teaching Assistant.