Win25 SciVis

TITLE: GPU-Accelerated Scientific Visualization Techniques (*GPU SciVis*)

Instructor: Dr. Colin Reinhardt, *ECE Affiliate Assistant Professor*

Overview:

In today's cloud-connected AI-driven cyberscape, the new problem is data overload. How can we make sense of all the tera/peta-bytes of data available at our fingertips? How do we extract useful engineering and scientific *information* from the vast raw datasets?

In many fields of engineering, medicine, and scientific research, interactive multidimensional computer visualization techniques can provide crucial insight. Scientific data visualization (SciVis) is a hybrid field at the cutting-edge intersection of real-time interactive computer graphics, physics-based simulations, machine learning, parallel software algorithms, and high-performance heterogeneous and GPU parallel computing hardware architectures.

Course Outline:

In this course, we'll learn the fundamentals of 3D computer graphics and development of interactive 3D visualization tools. These are useful for a variety of applications which generate large-scale complex multidimensional datasets, such as CT/MRI biomedical imagery, fluid flows from computational fluid dynamics (CFD) simulations, convective and radiative heat transfer modeling for aerospace engineering, and lidar remote-sensing point-clouds (to name just a few). They are also the foundation for the current rapidly evolving fields of 3D neural modeling, reconstruction, and generation.

In this course we'll study the following topics:

- the fundamentals of scientific visualization: scalar, vector, and volumetric field visualization methods, data representations, mesh and grid generation, sampling and interpolation, isosurface extraction, and basic color-space concepts.
- design and implementation of fundamental SciVis volumetric algorithms such as marching cubes, ray casting, splatting, and direct volume rendering (DVR).
- programming optimized SciVis algorithms using the parallelized GPU graphics pipeline (OpenGL 4.x) and compute kernel APIs and shaders.
- GPU hardware internals and how to design our programs to take optimal advantage of the host-device data interface, GPU cache and memory architecture, 2D/3D texture units, SPMD execution model.
- Domain-modeling and visualization techniques for point-cloud scattered datasets: radial basis functions, grid construction, triangulation, surface reconstruction methods.
- Discussions of how techniques studied relate to current important 3D neural/machine learning methods.

Lectures Plan

- L1: Course overview & logistics. AWS AppStream VDE usage, Computer Graphics (CG) fundamentals, GPU architecture overview, and modern OpenGL and GL Shading Language (GLSL) introductions.
- L2: CG, OpenGL, and GLSL continued: homogeneous coordinates, model, view, projection transformations
- L3: CG, OpenGL, and GLSL continued: lighting/shading/reflectance, advanced shaders
- L4: Textures and texture mapping, shader debugging techniques
- L5: SciVis: data representation, interpolation, (re)sampling, isosurface extraction & marching squares algorithm
- L6: Indirect volume rendering (IVR) 3D marching cubes algorithm, colormapping
- L7: Direct volume rendering (DVR) volume visualization techniques: ray-casting/marching, blending and transparency
- L8: (IVR/DVR) Volume vis (cont.)
- L9: Point cloud datasets: data management and visualization techniques
- L10: Point cloud datasets (cont.), other SciVis application examples

Homework DUE Dates

- HW1 due 1/17/25 by 11:59 PM (10pts)
- HW2 due 1/26/25 by 11:59 PM (10pts)
- HW3 due 2/3/25 by 11:59 PM (15 pts)
- HW4 due 2/16/25 by 11:59 PM (15 pts)
- HW5 due 3/3/25 by 11:59 PM (25 pts)
- HW6 due 3/19/25 by 11:59 PM (25 pts)

Prerequisites:

- This class will move quickly and assumes intermediate/advanced programming experience.
- Proficiency with C/C++ is strongly recommended.
- Familiarity with 3D computer graphics principles suggested.
- Familiarity with modern microprocessor and computer architecture suggested.

Course Structure & Grading:

- The course will be delivered via weekly live Zoom lecture (TUES 6-830 PM pacific time) and supplemented with additional Panopto recordings.
- The class grade is based on 6 homeworks, which increase in complexity and value.

Late Homework Policy

Lose 1 pt if submitted late, and another pt each day until 5 days late, then you lose 2 more at 10 days late and 1 more per additional 5 days until you hit zero pts.

If you have a personal emergency or situation or a doctor's note (or some other legitimate extenuating circumstance) please contact the professor and you will be granted an extension.

AI Usage Policy

- You are allowed (and even encouraged) to use AI coding assistant tools, such as ChatGPT, CoPilot, Claude, Codex, Code Intelligence, etc., for your homework programming assignments.
- **However, it is still strictly prohibited to directly copy code from the Internet or from someone else**. Such actions will result in a score of zero and a report to the university.