King Abdullah University of Science and Technology

KAUST

AMCS / CS 247 – Scientific Visualization
Lecture 1: Introduction

Markus Hadwiger, KAUST

Lecture Overview

Goals

- Basics: Learn the most important techniques in scientific visualization
- Practice: Implement scalar and vector/flow field visualization techniques in OpenGL

Time and location

- Monday + Thursday, 9:00 – 10:30, Building 9, Room 3222.

Course webpage:

`http://faculty.kaust.edu.sa/sites/markushadwiger/Pages/CS247.aspx`

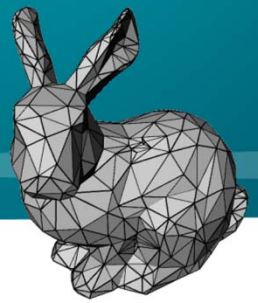
Contact

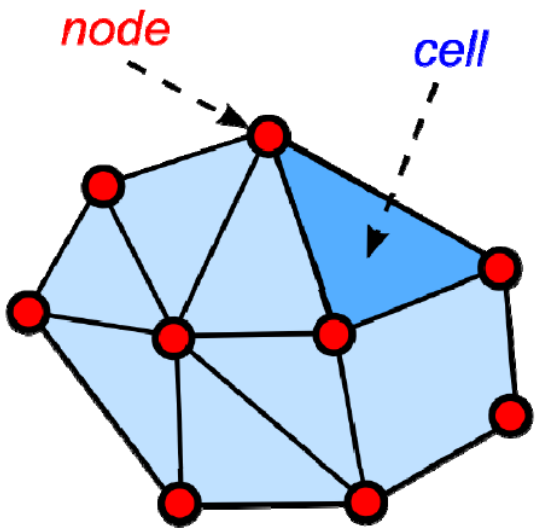
- Markus Hadwiger: **`markus.hadwiger@kaust.edu.sa`**
- Thomas Höllt: **`thomas.hollt@kaust.edu.sa`**
- Ali Awami: **`ali.awami@kaust.edu.sa`**

Prerequisites

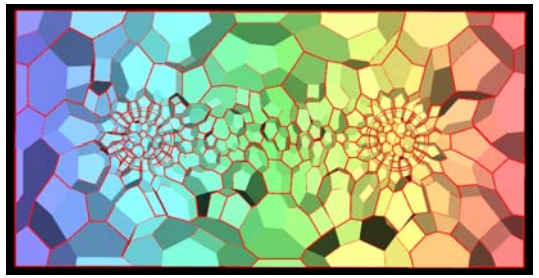
- **C/C++ programming**, computer graphics, linear algebra, basic calculus
- OpenGL experience very helpful !

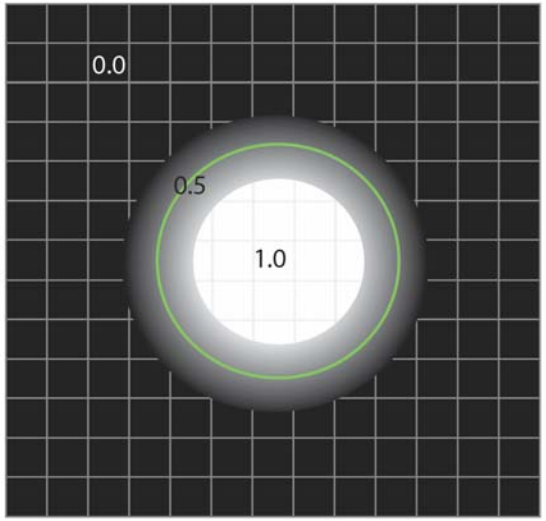
Syllabus (1)

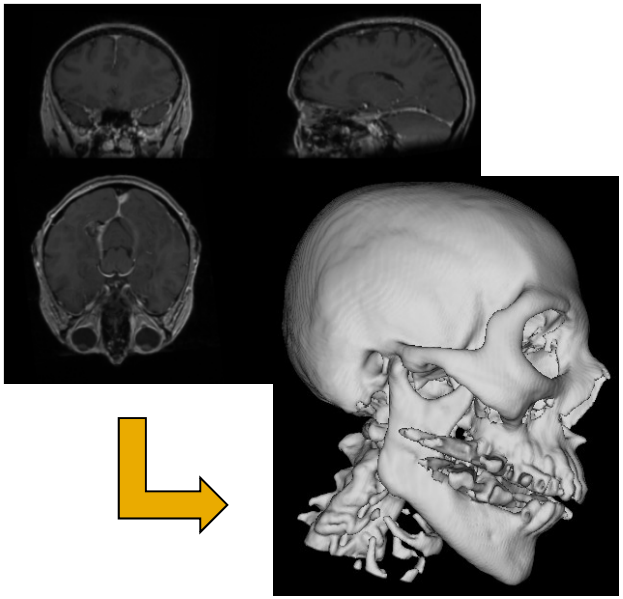


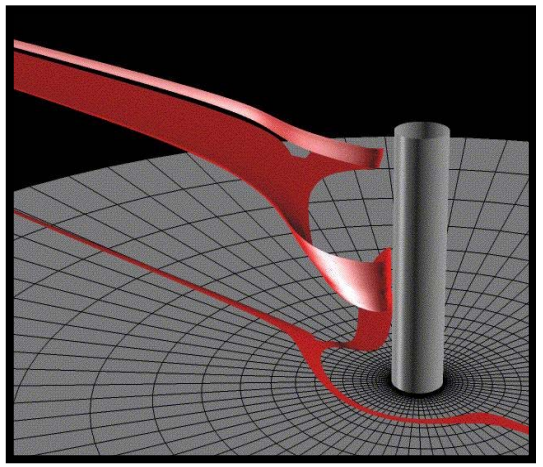


- Introduction
- Visualization basics, pipeline, and examples
- First scalar visualization example: iso-contouring
- GPU and computer graphics primer
- Data representation (grid types, data structures)



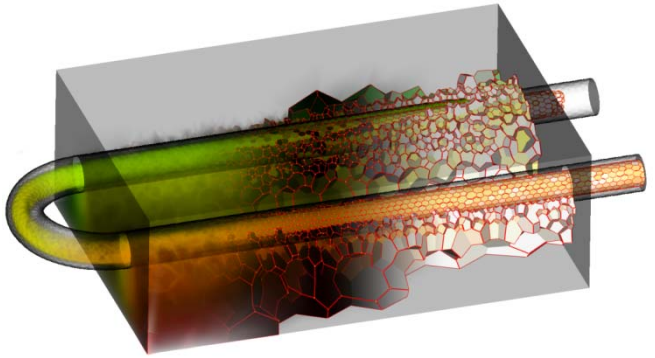






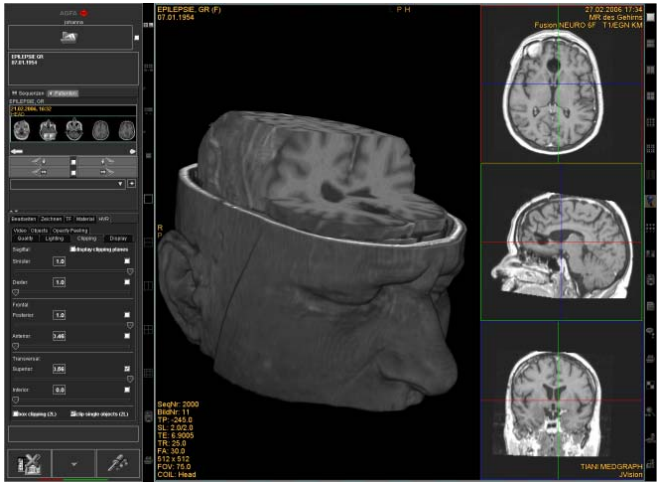
Syllabus (2)

- Scalar field visualization
  - Iso-surface rendering
  - Volume rendering
  - Transfer functions
  - Volume lighting
  - Unstructured grid visualization

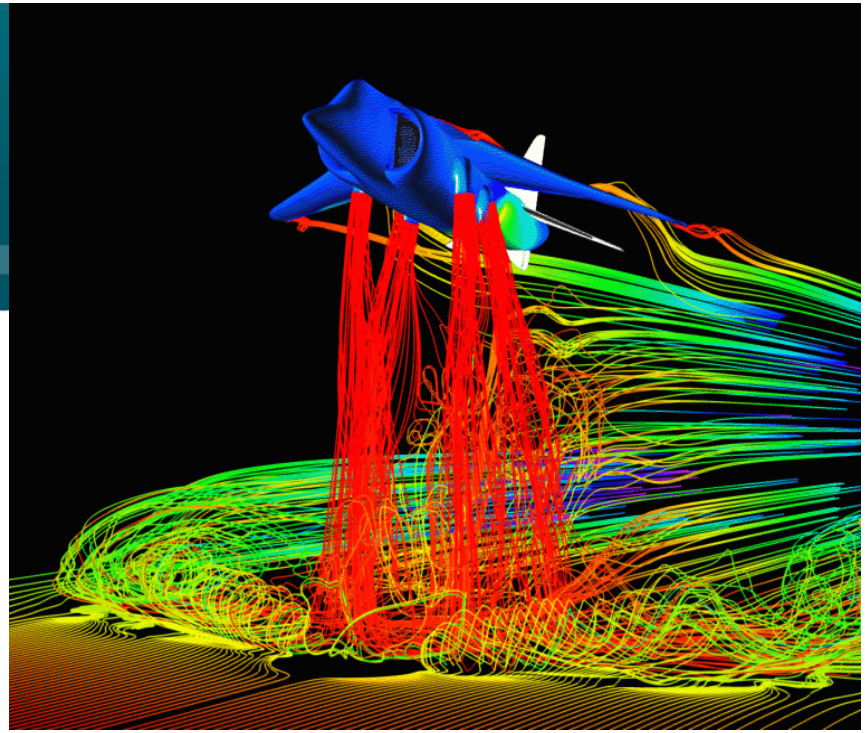


Applications

- Medical visualization
- Industrial CT (computed tomography)
- CFD (computational fluid dynamics) visualization of scalar quantities



Syllabus (3)



- Vector field and flow visualization
  - Direct vs. indirect techniques
  - Particle tracing
  - Integral curves and surfaces
  - Dense flow visualization techniques

Applications

  - CFD flow visualization
  - Weather visualization

- Basic tensor visualization
- Visualization systems

Lecture Structure and Grading

Lectures

Weekly reading assignments (required + optional)

- Part of quiz questions (see below)

Programming assignments

- 5 programming assignments; short written report + personal presentation for each

Quizzes

- 4 quizzes, 30 min each;
  announced a week in advance, roughly every 3-4 weeks
- From lectures, (required) reading assignments, programming assignments

Grading: 60% programming assignments; 40% quizzes

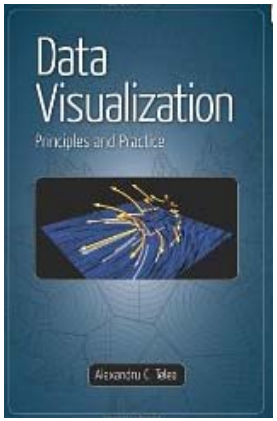
No mid-term/final exam!

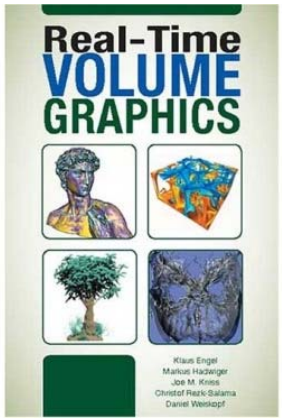
Resources

Course webpage:
`http://faculty.kaust.edu.sa/sites/markushadwiger/Pages/CS247.aspx`

Textbooks:

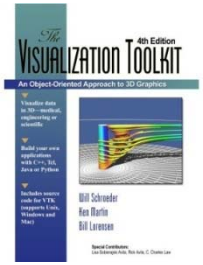
- Data Visualization: Principles and Practice
- Real-Time Volume Graphics



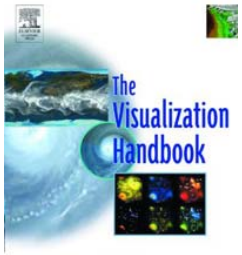


Additional books:

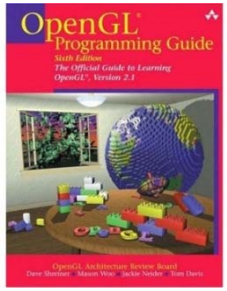
- The Visualization Toolkit:
  An Object-Oriented Approach
  to 3D Graphics (4th Edition)



- The Visualization Handbook



- OpenGL Programming Guide (8th edition, OpenGL 4.3)
  `www.opengl.org/documentation/red_book/`



For GPU, GPGPU, and graphics programming, also look here:

`http://faculty.kaust.edu.sa/sites/markushadwiger/Pages/CS380.aspx`

Programming Assignments (1)

5 assignments

- Based on C/C++ and OpenGL

  (if you know CUDA or OpenCL, you can also use it)

Organization

1. Use *bitbucket/git* to get material and submit solution
2. Get assignment info and framework by forking the git project
3. Separate Q&A session for each assignment (attendance optional)
4. Submit solution and report via git by submission deadline (Sundays)
5. Personal presentation after submission (Tuesday afternoon after deadline)

Programming Assignments (2)

- Submit via *bitbucket+git* at the latest on day the assignment is due (code, libs, everything that is needed to run your program)

- Submission must include short report (2 pages, pdf), including short explanation of algorithms, your solution, problems, how to run it, screenshots

- Personal presentations (Tuesday after submission), present your program live and explain source code (10-15 min)
  - Sign up for presentation slot in advance; sign-up sheet at Thomas's desk, Bldg 1, Room 2109
  - Use your own laptop (preferred!) or test on lab machine to guarantee it runs!

Programming Assignments (3)

Grading

- Submission complete, code working for all the required features
- Documentation complete (report, but also source code comments!)
- Personal presentation
- Optional features, coding style, clean solution
- Every day of late submission reduces points by 10%
- No direct copies from the Internet!
  You have to understand what you program:
  your explanations during the presentations will be part of the grade!

Programming Assignments (4)

Teaching Assistants:

- Thomas Höllt (**`thomas.hollt@kaust.edu.sa`**) – main contact for questions on programming assignments

  Office hours: Mondays from 17:00 – 18:00,
  Bldg 1, office 2109 (sea front)

- Ali Awami (**`ali.awami@kaust.edu.sa`**) – assisting in programming-related questions

  Send email or come by (open lab space, close to room 2118)

Help in programming assignments (in this order!):

1. **Think about it, read about it, google it!**
2. **Ask other students! (it especially helps to discuss major problems in groups!)**
3. Ask TAs (Thomas and Ali)

Programming Assignments Schedule (tentative)

Lab sign-up

- Setup bitbucket account, fork repository — until Sep 9

Assignment 1:

- Volume slice viewer — due: Sep 15

Assignment 2:

- Iso-contours and iso-surface rendering — due: Sep 29

Assignment 3:

- Volume ray-casting — due: Oct 20

Assignment 4:

- Flow vis 1 (hedgehog plots, streamlines, pathlines) — due: Nov 10

Assignment 5:

- Flow vis 2 (LIC with color coding) — due: Nov 24

Q&A Sessions Schedule (tentative)

Usually Monday from 17:00 – 18:00, Bldg 1, Room 2114

Assignment 1:

- Volume slice viewer — Sep 9

Assignment 2:

- Iso-contours and iso-surface rendering — Sep 19 (Thursday)

Assignment 3:

- Volume ray-casting — Oct 7

Assignment 4:

- Flow vis 1 (hedgehog plots, streamlines) — Oct 28

Assignment 5:

- Flow vis 2 (LIC) — Nov 18

Reading Assignment #1 (until Sep. 9)

Read (required):

- Data Visualization book, Chapter 1
- Data Visualization book, Chapter 2 until 2.3 (inclusive)
- Download and look at:
  NIH/NSF Visualization Research Challenges report

  http://vgtc.org/wpmu/techcom/national-initiatives/nihnsf-visualization-research-challenges-report-january-2006

- Start familiarizing yourself with OpenGL if you do not know it !

What is Scientific Visualization? (1)

The use of computer graphics for the analysis and presentation of computed or measured scientific data

- Started in 1987 by the US National Science Foundation (NSF) in the "Visualization in Scientific Computing" report

  `http://www.evl.uic.edu/core.php?mod=4&type=3&indi=348`

- First IEEE Visualization conference 1990

- 2006 NIH/NSF Visualization Research Challenges Report, Chris Johnson et al.

  `http://vgtc.org/wpmu/techcom/national-initiatives/nihnsf-visualization-research-challenges-report-january-2006`

"The purpose of computing is insight, not numbers"
Richard Hamming, 1971

What is Scientific Visualization? (2)

Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science.

McCormick, B.H., T.A. DeFanti, M.D. Brown,
Visualization in Scientific Computing,
Computer Graphics 21(6), November 1987

What is Scientific Visualization? (3)

The standard argument to promote scientific visualization is that today's researchers must consume ever higher volumes of numbers that gush, as if from a fire hose, out of supercomputer simulations or high-powered scientific instruments. If researchers try to read the data, usually presented as vast numeric matrices, they will take in the information at snail's pace. If the information is rendered graphically, however, they can assimilate it at a much faster rate.

R.M. Friedhoff and T. Kiely,
The Eye of the Beholder,
Computer Graphics World 13(8), pp. 46-, August 1990

What is Scientific Visualization? (4)

The use of computer imaging technology as a tool for comprehending data obtained by simulation or physical measurement by integration of older technologies, including computer graphics, image processing, computer vision, computer-aided design, geometric modeling, approximation theory, perceptual psychology, and user interface studies.

R.B. Haber and D. A. McNabb,

Visualization Idioms: A Conceptual Model for Scientific Visualization Systems,
Visualization in Scientific Computing,
IEEE Computer Society Press 1990.

What is Scientific Visualization? (5)

Scientific Visualization is concerned with exploring data and information in such a way as to gain understanding and insight into the data. The goal of scientific visualization is to promote a deeper level of understanding of the data under investigation and to foster new insight into the underlying processes, relying on the humans' powerful ability to visualize. In a number of instances, the tools and techniques of visualization have been used to analyze and display large volumes of, often time-varying, multidimensional data in such a way as to allow the user to extract significant features and results quickly and easily.

K.W. Brodlie, L.A. Carpenter, R.A. Earnshaw, J.R. Gallop, R.J. Hubbard, A.M. Mumford, C.D. Osland, P. Quarendon,
Scientific Visualization, Techniques and Applications,
Springer-Verlag, 1992.

What is Scientific Visualization? (6)

Scientific data visualization supports scientists and relations,

to prove or disprove hypotheses, and discover new phenomena

using graphical techniques.

The primary objective in data visualization is to gain insight into

an information space by mapping data onto graphical primitives.

H. Senay and E. Ignatius,
A Knowledge-Based System for Visualization Design,
IEEE Computer Graphics and Applications, pp. 36-47, November 1994

Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama