

# **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>

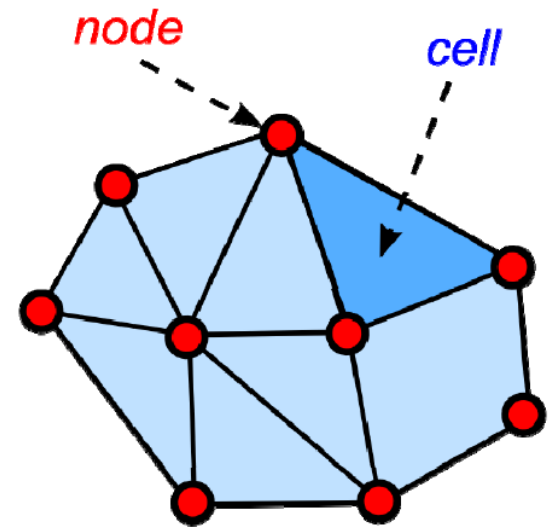
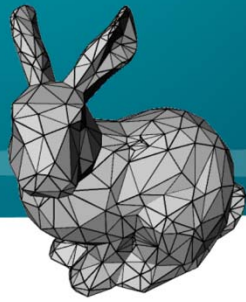
## Contact

- Markus Hadwiger: `markus.hadwiger@kaust.edu.sa`
- Thomas Theussl: `thomas.theussl@kaust.edu.sa`
- Ali Awami: `ali.awami@kaust.edu.sa`

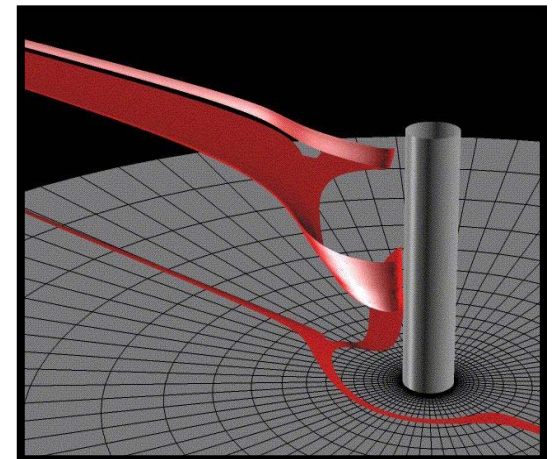
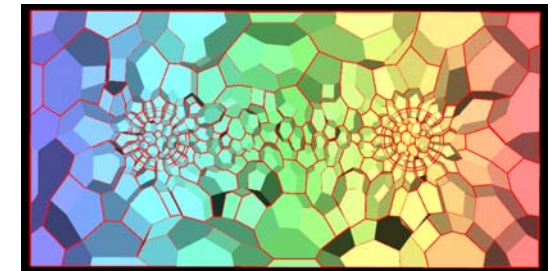
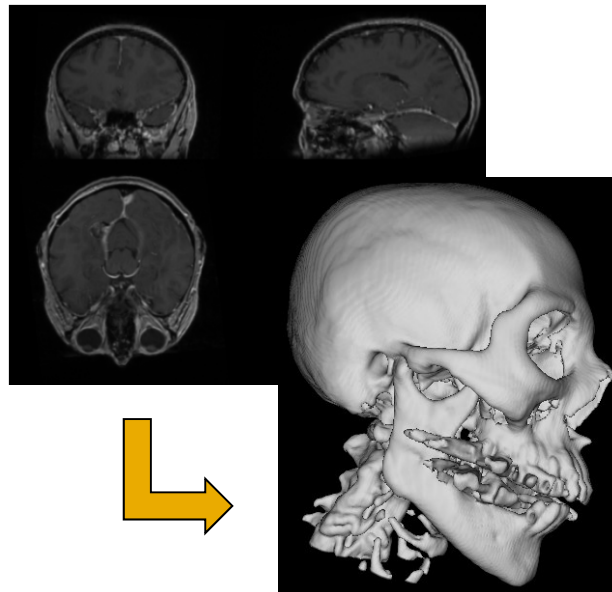
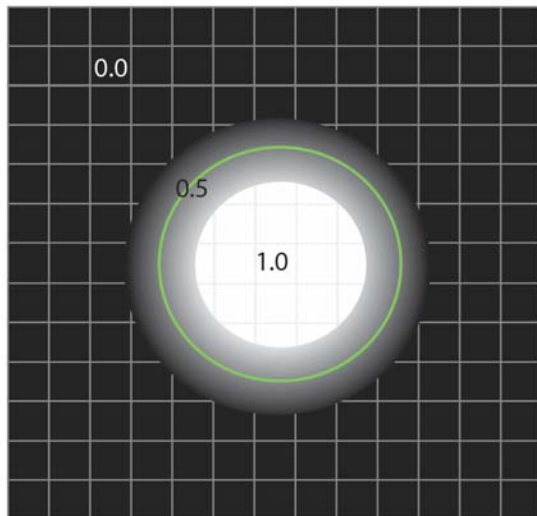
## Prerequisites

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

# Syllabus (1)

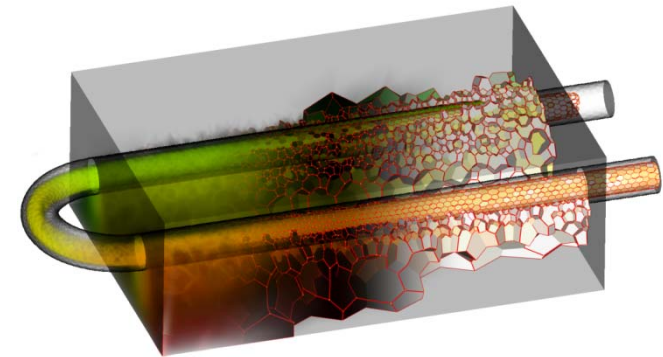
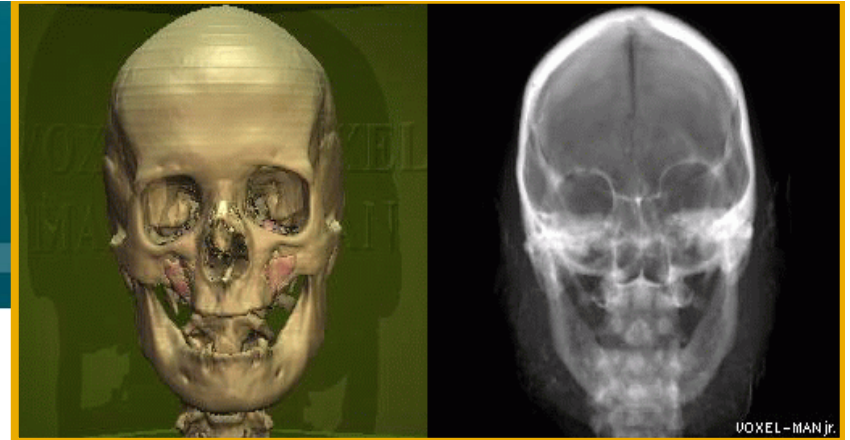


- Introduction
- Visualization basics, pipeline, and examples
- First scalar visualization example: iso-contouring
- GPU and computer graphics primer
- Data representation (grid types, data structures)



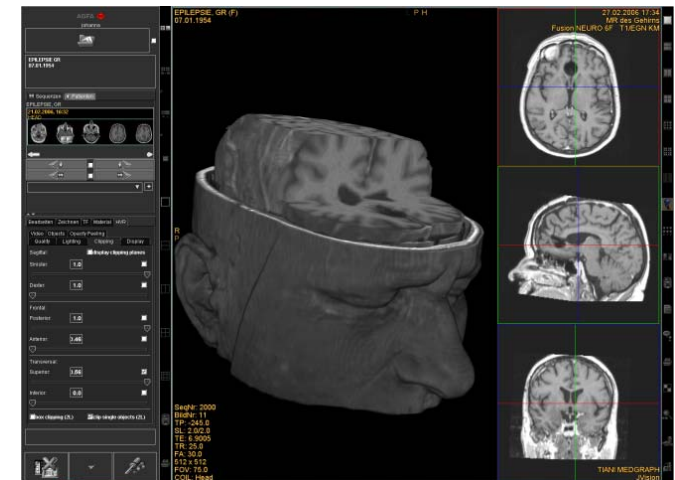
# Syllabus (2)

- 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



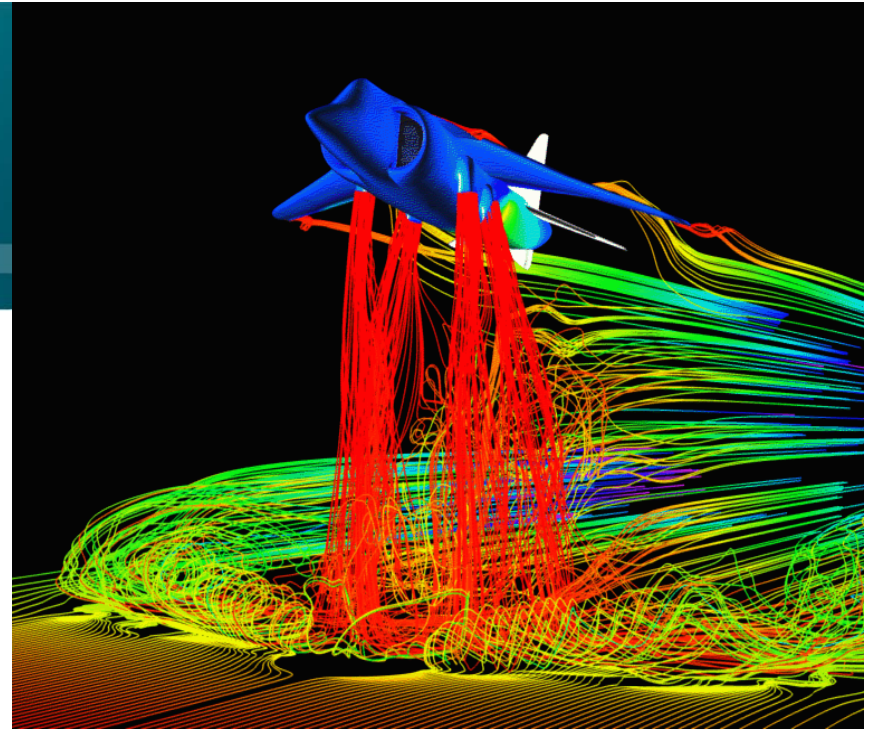


# Syllabus (3)

- 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:

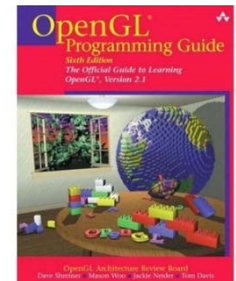
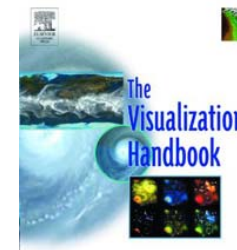
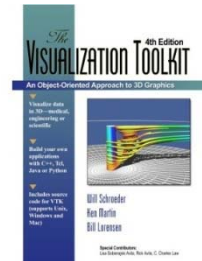
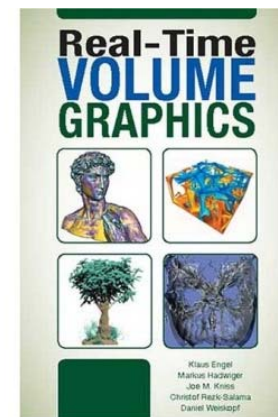
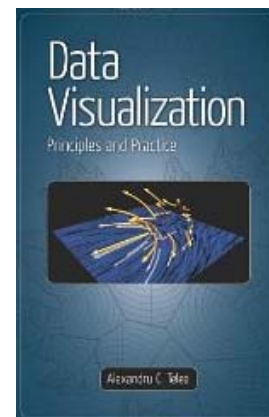
<http://faculty.kaust.edu.sa/sites/markushadwiger/Pages/CS247.aspx>

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 (8<sup>th</sup> edition, OpenGL 4.3)  
[www.opengl.org/documentation/red\\_book/](http://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  
(`bitbucket.org/kaust_cs247/cs247_2014`)
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 my office door, Bldg 1, Room 2119
  - 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 Theussl ([thomas.theussl@kaust.edu.sa](mailto:thomas.theussl@kaust.edu.sa)) – main contact for questions on programming assignments

Open lab space, Bldg 1, Room 2101; (close to room 2119)



- Ali Awami ([ali.awami@kaust.edu.sa](mailto:ali.awami@kaust.edu.sa)) – assisting in programming-related questions

Open lab space, Bldg 1, Room 2101; (close to room 2119)



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 Aug 24

## Assignment 1:

- Volume slice viewer due: Aug 31

## Assignment 2:

- Iso-contours and iso-surface rendering due: Sep 21

## Assignment 3:

- Volume ray-casting due: Oct 19

## Assignment 4:

- Flow vis 1 (hedgehog plots, streamlines, pathlines) due: Nov 9

## Assignment 5:

- Flow vis 2 (LIC with color coding) due: Nov 30

# Q&A Sessions Schedule (tentative)



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

## Assignment 1:

- Volume slice viewer Aug 24 (Sun)

## Assignment 2:

- Iso-contours and iso-surface rendering Sep 9 (Tue)

## Assignment 3:

- Volume ray-casting Sep 30 (Tue)

## Assignment 4:

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

## Assignment 5:

- Flow vis 2 (LIC) Nov 18 (Tue)



# Reading Assignment #1 (until Aug 25)



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?



**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*

# Thank you.

Thanks for material

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