

AMCS / CS 247 – Scientific Visualization

Lecture 2: Introduction, Pt. 2

Markus Hadwiger, KAUST

Reading Assignment #1 (until Aug 31)



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://tab.computer.org/vgtc/vrc/
NIH-NSF-VRC-Report-Final.pdf`

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

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:

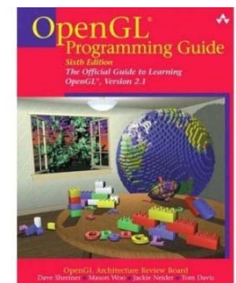
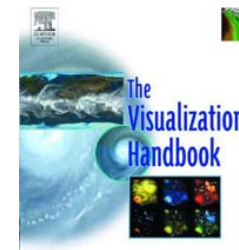
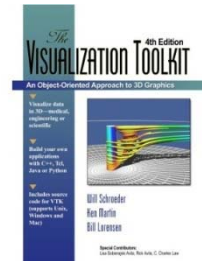
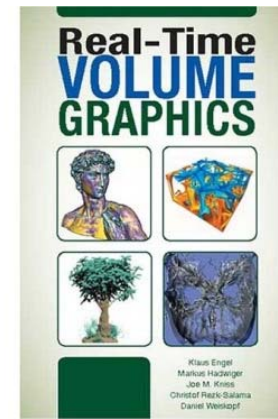
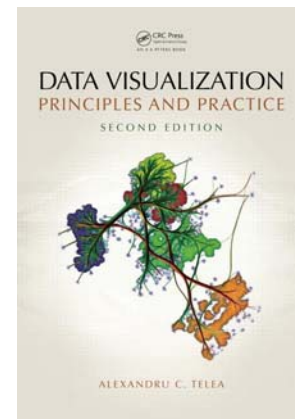
<http://faculty.kaust.edu.sa/sites/markushadwiger/Pages/CS247.aspx>

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
(`bitbucket.org/kaust_cs247/cs247_2015`)
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
 - 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 Assignment #1



Lab sign-up

- Setup bitbucket account, fork repository until Aug 30
- You can a git client, e.g., from <http://git-scm.com/downloads>

Q&A assignment 1:

- Aug 30 (Sun) 17:00 – 18:00, Bldg 1, Room 2114

Deadline assignment 1:

- Volume slice viewer due: Sep 6

Programming Assignment #1: Slice Viewer



Basic tasks

- Load volume data
- Specify volume data as OpenGL 3D texture
- Display slice

Minimum

- Show one view, for the selected major axis
- Selectable: major axis, slice position
- Make sure the aspect ratio is correct

Bonus

- Show all three views simultaneously
- Allow arbitrary slice plane orientations

Programming Assignment #1 Example



```
#include <iostream>

G:\Development\git\Teaching\Work\CS247_Assignment1\64\Debug\CS247_Assignment1.exe
GL_VERSION major=4 minor=3
Keyboard commands:
b - Toggle among background clear colors
w - Increase current slice
s - Decrease current slice
a - Toggle viewing axis
1 - Load lobster dataset
2 - Load head dataset
3 - Load hydrogen dataset
loading data ../Datasets/skewed_head.dat
volume dimensions: x: 184, y: 256, z:170
downloading volume to 3D texture
increasing current slice: 86
increasing current slice: 87
increasing current slice: 88
increasing current slice: 89
increasing current slice: 90

int printOpenGLError(char *file, int line)
{
    // Returns 1 if an OpenGL error occurred, 0 otherwise.
    GLenum glErr;
    int retCode = 0;

    glErr = glGetError();
    while (glErr != GL_NO_ERROR)
    {
        printf("glError in file %s @ line %d: %s\n", file, line, gluError
        retCode = 1;
        glErr = glGetError();
    }
    return retCode;
}

#define printOpenGLError() printOpenGLError(__FILE__, __LINE__)
```



Programming Assignment #1 Example



```
G:\Development\git\Teaching\Work\CS247_Assignment1\64\Debug\CS247_Assignment1.exe
b - Toggle among background clear colors
w - Increase current slice
s - Decrease current slice
a - Toggle viewing axis
1 - Load lobster dataset
2 - Load head dataset
3 - Load hydrogen dataset
loading data ../Datasets/skewed_head.dat
volume dimensions: x: 184, y: 256, z:170
downloading volume to 3D texture
increasing current slice: 86
increasing current slice: 87
increasing current slice: 88
increasing current slice: 89
increasing current slice: 90
toggling viewing axis to: 0
increasing current slice: 93
increasing current slice: 94
increasing current slice: 95
toggling viewing axis to: 1
decreasing current slice: 127
decreasing current slice: 126
decreasing current slice: 125
decreasing current slice: 124
```

```
int printOpenGLError(char *file, int line)
{
    // Returns 1 if an OpenGL error occurred, 0 otherwise.
    GLenum glErr;
    int retCode = 0;

    glErr = glGetError();
    while (glErr != GL_NO_ERROR)
    {
        printf("glError in file %s @ line %d: %s\n", file, line, gluErrorString(glErr));
        retCode = 1;
        glErr = glGetError();
    }
    return retCode;
}
```



Programming Assignment #1 Example



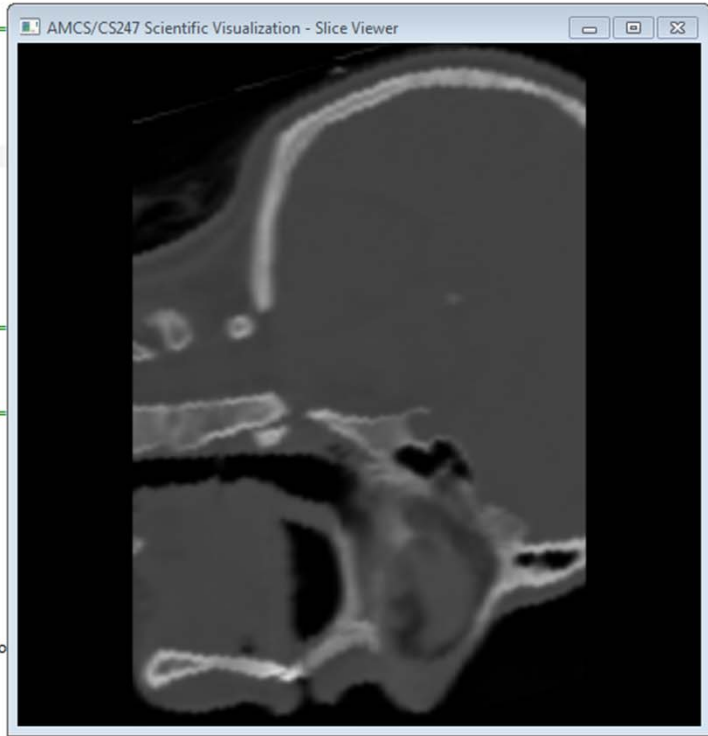
```
#include <iostream>

G:\Development\git\Teaching\Work\CS247_Assignment1\Debug\CS247_Assignment1.exe
GL_VERSION major=4 minor=3

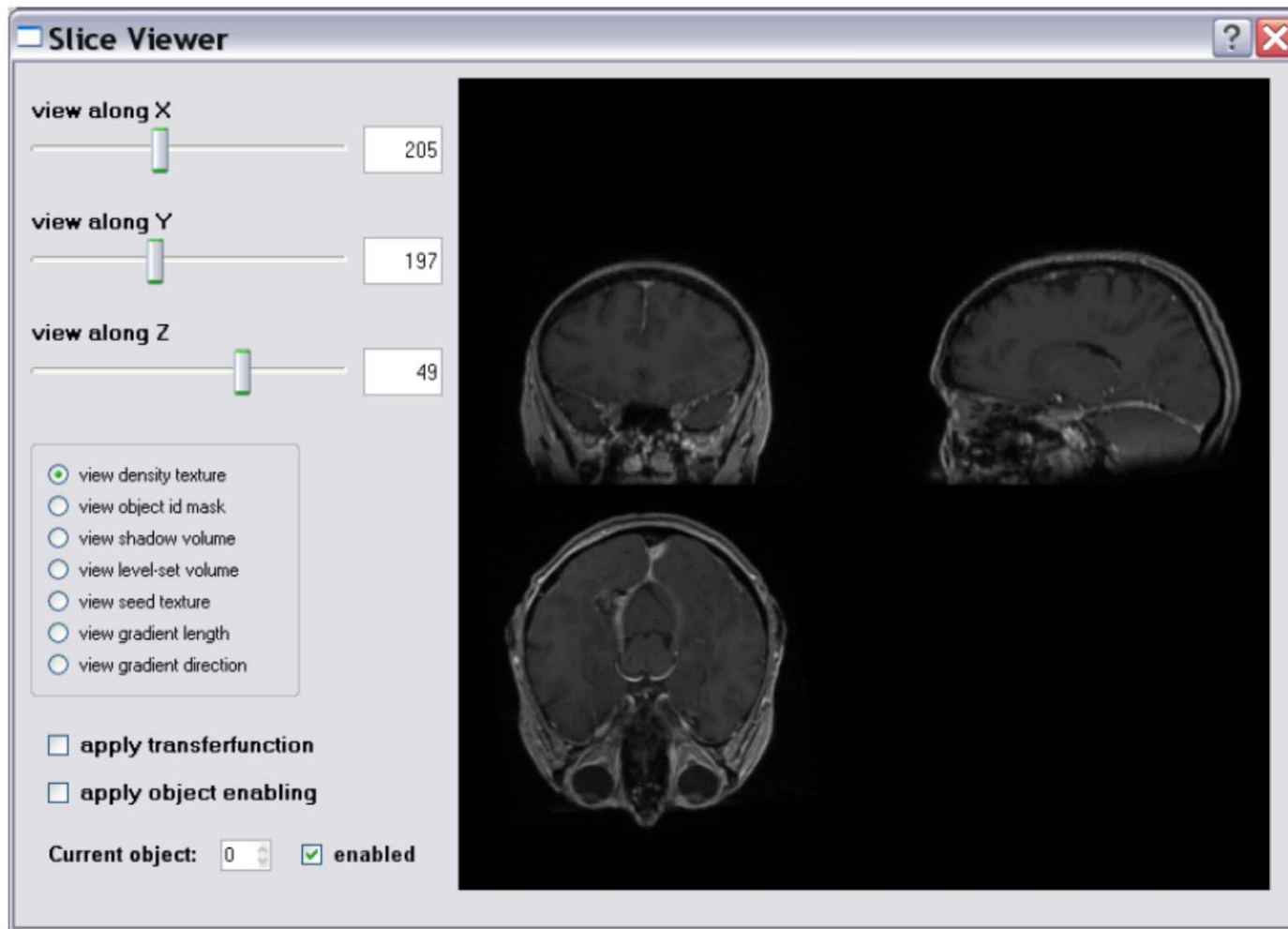
Keyboard commands:
b - Toggle among background clear colors
w - Increase current slice
s - Decrease current slice
a - Toggle viewing axis
1 - Load lobster dataset
2 - Load head dataset
3 - Load hydrogen dataset
loading data ../Datasets/skewed_head.dat
volume dimensions: x: 184, y: 256, z:170
downloading volume to 3D texture
increasing current slice: 86
increasing current slice: 87
increasing current slice: 88
increasing current slice: 89
increasing current slice: 90
toggling viewing axis to: 0
increasing current slice: 93
increasing current slice: 94
increasing current slice: 95

int printOpenGLError(char* file, int line)
{
    // Returns 1 if an OpenGL error occurred, 0 otherwise.
    GLenum glErr;
    int retCode = 0;

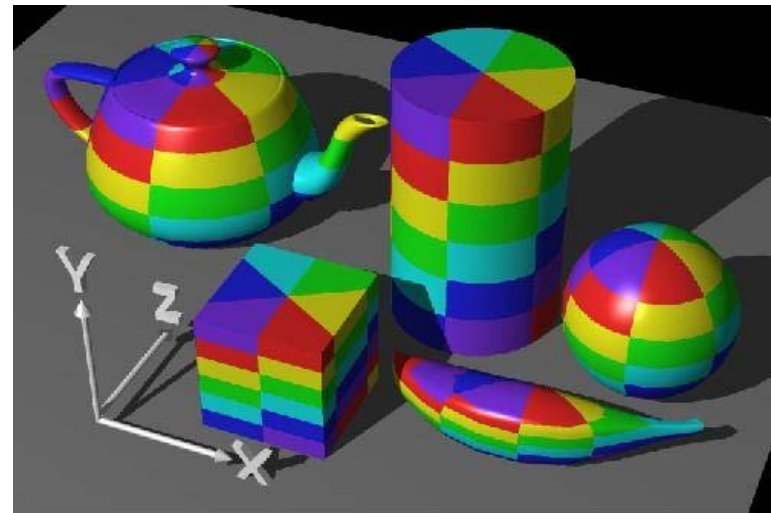
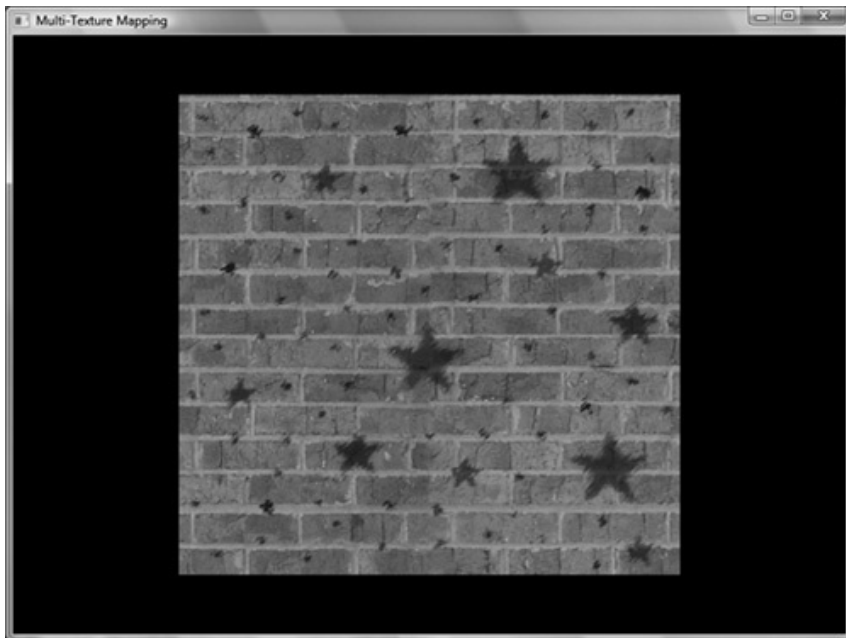
    glErr = glGetError();
    while (glErr != GL_NO_ERROR)
    {
        printf("glError in file %s @ line %d: %s\n", file, line, gluError
        retCode = 1;
        glErr = glGetError();
    }
    return retCode;
}
```



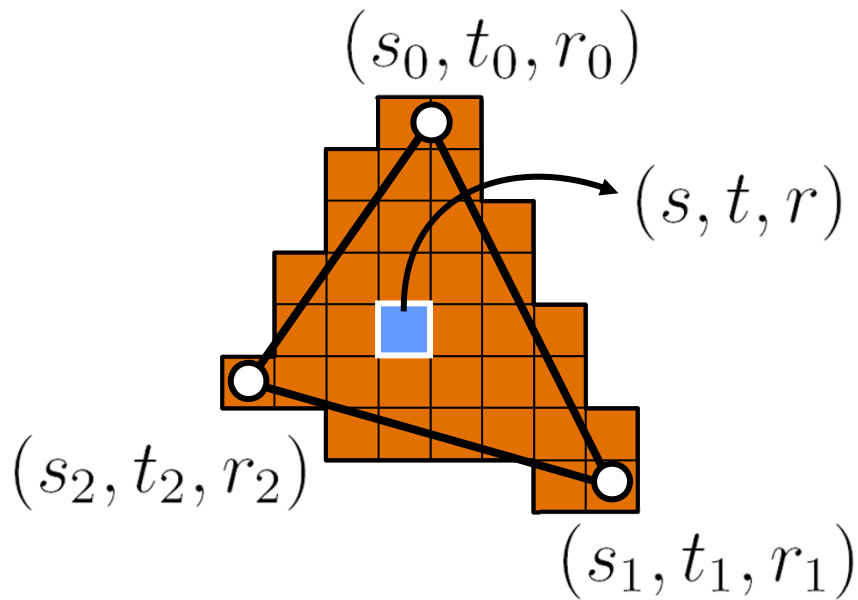
Programming Assignment #1 Example



Texture Mapping



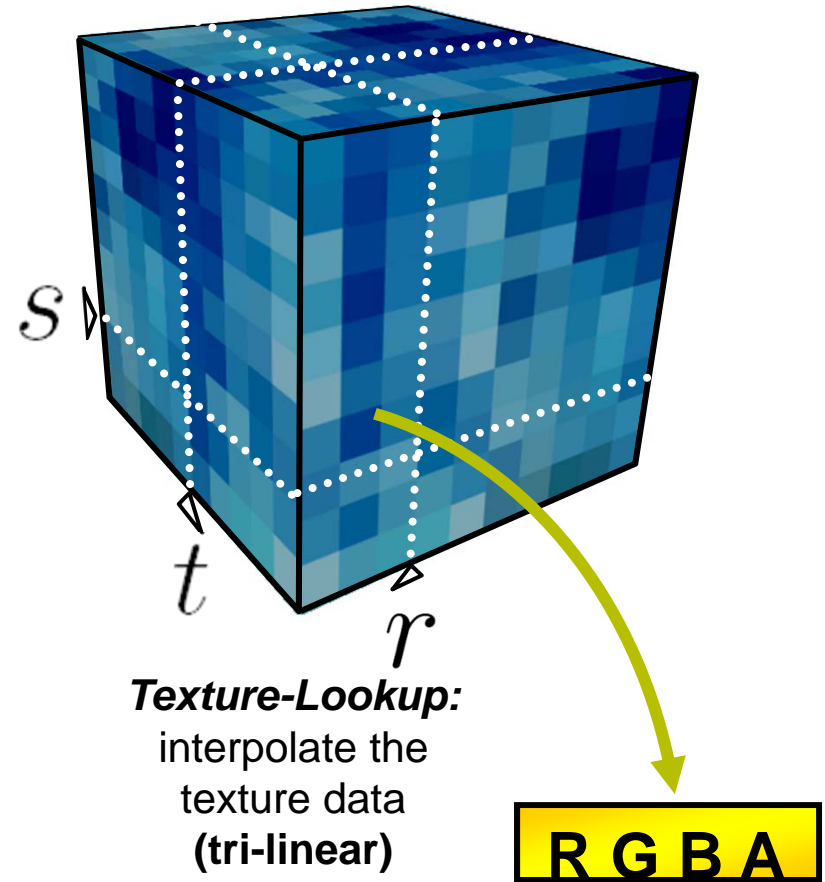
3D Texture Mapping



For each fragment:
interpolate the
texture coordinates
(barycentric)

Or:

Use arbitrary, computed coordinates



Nearest-neighbor for "array lookup"

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://tab.computer.org/vgtc/vrc/
NIH-NSF-VRC-Report-Final.pdf](http://tab.computer.org/vgtc/vrc/NIH-NSF-VRC-Report-Final.pdf)

“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