180.765 Seminar Wissenschaftliches Arbeiten

WS 2017/18

Organizer: Peter Mindek, Hsiang-Yun Wu
Teaching staff: Tobias Klein, Haichao Miao, David Kouřil, Manuela Waldner, Renata Raidou, Ivan Viola, Eduard Gröller

Institute of Computer Graphics and Algorithms
Vienna University of Technology
Seminar

- Get an idea how scientific work is carried out (in Visualization / CG)
  - Practice to review literature and get familiar with a particular scientific topic
    - Selecting, reading and understanding
    - Summarizing and explaining (orally and written)
    - Comparing and discussing
  - Practice to give a talk
  - Active discussion participation
Seminar - Procedure

1. Select a topic
Select a Topic

- Topics will be presented today
- Topic abstracts are available at
  http://cg.tuwien.ac.at/courses/WissArbeiten/index.html
- Alone or in groups of 2
- Choose a topic and wait until it is confirmed
  mindek@cg.tuwien.ac.at

Topic Selection Deadline: 20.10.2017
Seminar - Procedure

1. Select a topic
2. **Submit a literature list**
Submit a Literature List

- Meeting with Supervisor
- List of papers related to the topic

- Literature List Deadline: 27.10.2017
Seminar - Procedure

1. Select a topic
2. Submit a literature list
3. Attend 3 lectures
Attend 3 Lectures

- **10.11.2017, 14:00 - 16:00**
  Wie schreibt man eine wissenschaftliche Arbeit
  *Professor Wimmer*

- **23.11.2017, 13:00 - 15:00**
  Forschung und wie sie funktioniert
  *Professor Gröller*

- **30.11.2017, 13:00 - 15:00**
  Wie halte ich einen Vortrag
  *Professor Purgathofer*
Seminar - Procedure

1. Select a topic
2. Submit a literature list
3. Attend 3 lectures
4. **Write a report**
Write a Report

- State-of-the-Art Report
- Final Report: 6-8 pages (12-16 pages / group)
- In English
- Format as for a scientific paper
  - \LaTeX{} (Template on the webpage)
- Regular Meetings with Supervisor

Deadline Final Version Report: 12.1.2018
Seminar - Procedure

1. Select a topic
2. Submit a literature list
3. Attend 3 lectures
4. Write a report
5. Give a presentation
Give a Presentation

- Use institute’s PowerPoint template for presentations (template is on the webpage)
- In English
- 15 + 3 minutes
- Active discussion participation

Presentation Day: 20.12.2017

In case of too many students, an additional presentation day will be announced and/or the length of the presentation will be adjusted. This will be communicated in advance.
Seminar - Procedure

1. Select a topic
2. Submit a literature list
3. Attend 3 lectures
4. Write a report
5. Give a presentation
Grading

- Two parts
  - 1\textsuperscript{st} (central) part: 15\% of the grade
  - 2\textsuperscript{nd} part: 85\% of the grade
  - It is necessary to attend the 3 lectures to get a positive grade!

- Grading criteria:
  - 50\% written report
  - 40\% presentation
  - 5\% attendance during the presentations
  - 5\% active discussion after the presentations
Report Grading

- Grading criteria:
  - Structure, figures,...
  - Language
  - Content
  - References

- Points will be deducted for:
  - Delayed submission
  - Page number below 6 (12)

- Plagiarism check!
“plagiarism involves the use of another person's work without full and clear referencing and acknowledgement”

http://www.usq.edu.au/library/referencing/what-is-plagiarism
Presentation Grading

Grading Criteria
- Content Expertise
- Didactic / Preparation
- Presentation Technique
- Overtime
Important Dates

- **20.10.2017**: Select your topic
- **27.10.2017**: Submit your literature list
- Attend 3 lectures (in ICGA seminar room):
  - **10.11.2017, 14:00**: Wie schreibt man eine wissenschaftliche Arbeit
  - **23.11.2017, 13:00**: Forschung und wie sie funktioniert
  - **30.11.2017, 13:00**: Wie halte ich einen Vortrag
- **8.12.2017**: Submit draft report
- **20.12.2017**: Talks (in seminar room)
- **12.01.2018**: Submit final report

All submissions to mindek@cg.tuwien.ac.at, wu@cg.tuwien.ac.at
Topics 2017/2018
1. Visualization of Search Results

- How to visually present a lot of text- and image-based query results

- Challenges:
  - Unstructured (mainly text) data
  - Heterogeneous data (text, images...)

Manuela Waldner
2. Virtual Reality Visualizations

- Explore applications, chances, and challenges when using virtual reality for visualizing spatial and

![Medical Visualization](image1)

![Flow Visualization](image2)

![Information Visualization](image3)
3. Mixed and Augmented Reality Visualizations

- Explore applications, chances, and challenges when using mixed / augmented reality for visualizing spatial and abstract data

Karlsruhe Institute of Technology  
TU Munchen  
www.loom.io

Flow Visualization  
Medical Visualization  
Information Visualization
What are the state-of-the-art techniques (both hardware and software) that enable Augmented Reality to be precise enough for regular consumer applications?
5. Rendering Systems Architectures

- Research the approaches to designing rendering systems used in i.e. video games.
6. Visual Analytics for (Bio-)Medical Applications

[Hoellt et al. 2014]
7. Progressive Visual Analytics

[Stolper et al. 2014]
8. Evaluation methods in Medical Visualization

[Glasser et al. 2016]
9. Network Visualization for Biological Pathways

Networks are well-known representations for describing a relationship of entities between data samples. Thus, it is also intuitive to use networks as a base for visually describing biological interactions.
Automatic map generation becomes an interesting topic because it can be used for game design, house design, as well as map design. The results of these approaches often serve as a base for the hand-drawn design.
11. Abstract Visualization of DNA

- Abstract representations to depict certain features of DNA
- From low level (atomic details) to high level (nucleotides and single strands)

Miao et al. 2018

Miao et al. 2018

Abrahamsson & Plotkin 2009

by Michael Ströck
12. 2D Navigation in 3D Space

- Using 2D navigation to explore data in 3D space

Fiber tracts
Jianu et al. 2009

LiveSync
Kohlmann et al. 2006
13. Position-Based Dynamics
14. Screen-Space Advanced Illumination Algorithms
15. Performance Visualization

```cpp
int floatCompare(const void *a, const void *b) {
    return *(const float *)a - *(const float *)b;
}

void bubblesort(std::vector<float> &v) {
    for (unsigned end = v.size() - 1; end >= 0; end--) {
        bool swapped = false;
        for (unsigned i = 0; i < end; i++) {
            if (v[i] > v[i + 1]) {
                std::swap(v[i], v[i + 1]);
                swapped = true;
            }
        }
        if (!swapped) {
            break;
        }
    }
}

int main(int argc, char *argv[]) {
    // Get command line arguments.
    std::string inputfile, sortalgo;
    unsigned random;
    bool print, printhistory;
    try {
        // Create command line parser.
        TCLAP::CmdLine cmd("Sort a file's worth of numbers according to some algorithm");
        TCLAP::ValueArg<std::string> inputfileArg("i", "Input-file", "Newline-separated numbers",
                                                  false, "", "filename", cmd);
        TCLAP::ValueArg<std::string> sortalgoArg("s", "Sort algorithm", "Any algorithm",
                                                   false, "", "sortalgo", cmd);
        TCLAP::ValueArg<bool> randomArg("r", "Randomize", "Randomize input data",
                                                         false, true, "false", cmd);
        TCLAP::ValueArg<bool> printArg("p", "Print", "Print sorted array",
                                             false, true, "false", cmd);
        TCLAP::ValueArg<bool> printhistoryArg("h", "Print history", "Print history of sort process",
                                              false, true, "false", cmd);
        cmd.parse(argc, argv);
        inputfile = inputfileArg.getValue();
        sortalgo = sortalgoArg.getValue();
        random = randomArg.getValue();
        print = printArg.getValue();
        printhistory = printhistoryArg.getValue();
        // Process the input file and sort.
        // Print the sorted array.
        // Print the history of the sort process.
    } catch (const TCLAP::NoOptionException &e) {
        cmd.handleError(e.getMessage());
    } catch (const TCLAP::OptionException &e) {
        cmd.handleError(e.getMessage());
    } catch (const TCLAP::ValueException &e) {
        cmd.handleError(e.getMessage());
    } catch (const std::exception &e) {
        cmd.handleError(e.what());
    }
    // Print the sorted array.
    // Print the history of the sort process.
}
```
16. DSLs in Visualization

- DSL = Domain-specific language
- How to incorporate domain knowledge in a language

<table>
<thead>
<tr>
<th>source code and libraries</th>
<th>expert programmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSLs</td>
<td>expert user</td>
</tr>
<tr>
<td>data flow modules</td>
<td>intermediate</td>
</tr>
<tr>
<td>turn-key systems</td>
<td>novice</td>
</tr>
</tbody>
</table>

**Program Editor**

```c
void updateRendering(float x) {
    // use slang renderer
    using renderer;
    focus = value > x;
    using;
}
```

**Interactive Interpretation and Visualization**

- `x = 0;`
- `x = 0.05;`
- `x = 0.1;`
- `x = 0.2;`
17. Multi-Scale Visualization
18. Special Effects in Computer Graphics
1. Visualization of Search Results
2. Virtual Reality Visualizations
3. Mixed and Augmented Reality Visualizations
4. Augmented Reality: State of Technology
5. Rendering Systems Architectures
6. Visual Analytics for (Bio-)Medical Applications
7. Progressive Visual Analytics
8. Evaluation methods in Medical Visualization
9. Network Visualization for Biological Pathways
10. Automatic Map Layout using Space Partitioning Techniques
11. Abstract Visualization of DNA
12. 2D Navigation in 3D Space
13. Position-Based Dynamics
14. Screen-Space Advanced Illumination Algorithms
15. Performance Visualization
16. DSLs in Visualization
17. Multi-Scale Visualization
18. Special Effects in Computer Graphics
1. Astronomical Visualization
2. Realistic Medical Rendering
3. Anatomical Landmarking
4. Artificial Neural Network Visualization
5. Computer Aided Design
6. Animated Transitions in Data Visualization
7. Integration of Spatial and Abstract Visualization
8. Saliency in Graphics and Visualization
9. Big Data Visual Analytics
10. Structure Preserving Projection
11. Extreme Scale In Situ Visualization
12. Text and Document Visualization
13. Security Visualization
14. Performance Visualization
15. DSLs in Visualization
16. Volume Editing & Annotation
17. Volume Manipulation
18. Implicit Surfaces
19. Visualization Systems
20. Illustrative Shading Techniques
21. Smart Visibility Techniques
22. Grammar-based Computer Graphics and Visualization
23. Molecular Visualization
24. Uncertainty Visualization
25. Parameter Visualization
26. Procedural Plant Modelling
27. Procedural Content Generation
A very short introduction to LaTeX

- Document markup language
  - “programming“ a text document
- Similarities to HTML
- No WYSIWYG
- Most convenient to use a LaTeX distribution and a LaTeX IDE (integrated development environment)
A very short introduction to LaTeX

First install a LaTeX Distribution
- MiKTeX (for Windows)

Then a LaTeX IDE
- TeXnicCenter
- Texmaker
- LEd
A very short introduction to TEXnicCenter

- Extract the archive `acmsiggraph.zip`
- In TEXnicCenter open `template.tex`
  1. Select LaTeX=>DVI=>PDF output profile
  2. Select LaTeX=>PS output profile
     - Convert PS to PDF using Adobe Distiller
  3. Select LaTeX=>PDF output profile (pdflatex)
     - For eps images use `\usepackage{epstopdf}`
A very short introduction to LaTeX

- Work with 2 files:
  - A .tex file for the text
  - A .bib file for the bibliography which is used by the citations command \cite
Questions?

http://www.cg.tuwien.ac.at/courses/WissArbeiten/index.html