Wissenschaftliches Arbeiten Part 2 with CG Group
193.052, SS 2021, 2.0h (3 ECTS)

Philipp Erler
https://www.cg.tuwien.ac.at/staff/PhilippErler.html

Research Division of Computer Graphics
Institute of Visual Computing & Human-Centered Technology
TU Wien, Austria

Organization

- Organization via TUWEL
  https://tuwel.tuwien.ac.at/course/view.php?id=36384
- General information on LVA site
  https://www.cg.tuwien.ac.at/courses/WissArbeiten/SE/2021S
- Topics and organization are presented today
- Topic selection in TUWEL -> fixed by end of the week

Goals

- Practice selecting, reading and understanding
  - Search and select papers relevant to your topic
  - Summarize them as a state-of-the-art report
  - Prepare a talk about your topic in the seminar

- This permits in-depth familiarization with the topic

Tasks

- Submit a literature list
  - Chosen with supervisor
- Attend 3 lectures
- Meetings with supervisor
  - paper selection
  - discussion of papers
  - preparing talk slides
- Alternative: evaluate and compare algorithms
- Final presentation in seminar

Literature List

- Analyze recent papers (select with supervisor)
- Study secondary literature to understand topic
- How to find relevant papers:
  - Google Scholar: key words and operators
  - Digital libraries: IEEE, ACM, …
  - Survey papers, often-referenced papers
  - Skim the papers at least
- Submit a list of ca. 10 papers in TUWEL
  e.g. 8 technical papers + 2 survey papers or text books
  → **official registration** (CG seminar)

State-of-the-Art Report (STAR)

- LaTeX template
  - Information on course website
  - Overleaf reference project available to copy
- Submit the paper in PDF format in TUWEL
- **First submission must be complete**
  - Min. 8 pages, preferably in English
  - All papers mentioned and complete structure
  - This version will be reviewed but not graded
- Start early! Plan at least 4 weeks for reading and writing.

Scientific Review

- You will get the first submission of another student to review
- Typical conference review form (Eurographics)
- This helps author to improve the manuscript
- Guides on review writing on course website
- You will receive 2 reviews (student, supervisor)
- Improve final report (camera-ready submission) according to reviews
- Plagiates -> Fail!
  Institute Guidelines

Seminar Presentations

- Duration:
  - depending on number of students
  - 10-15 minutes presentation +3-5 minutes discussion
- Presentation (preferably in English)
  - Prepare slides in advance, using template
  - Focus is on overview/comparison of methods
  - Present so that other students will understand it
  - ~~Submitted slides are presented on seminar PC~~ via Zoom
  - Recorded video + live Q&A
- Active discussion is mandatory and graded

Dates

- Rough overview, see TUWEL course for details
- 3 weeks for meeting supervisor and literature list
- 7 weeks for report
  - 3 lectures of 2h during this time
  - Start early!
- 1 week for reviews
- 2 weeks for presentation preparation and final paper

Grading

- Grades: 1: >88%, 2: 75%, 3: 63%, 4: 50%, 5: <50%
- Every submission must be 4 or better, otherwise 5 overall
- Late submission:
  - -1 point per started hour
    -> fail course after 50h
  - You will delay the next task for everyone!

Task	Points
Lecture attendance	+5
Review	+15
Presentation	+30
Participation in discussion	+5
Final report	+45
Late	-x
Sum	=100-x

Topic Presentation

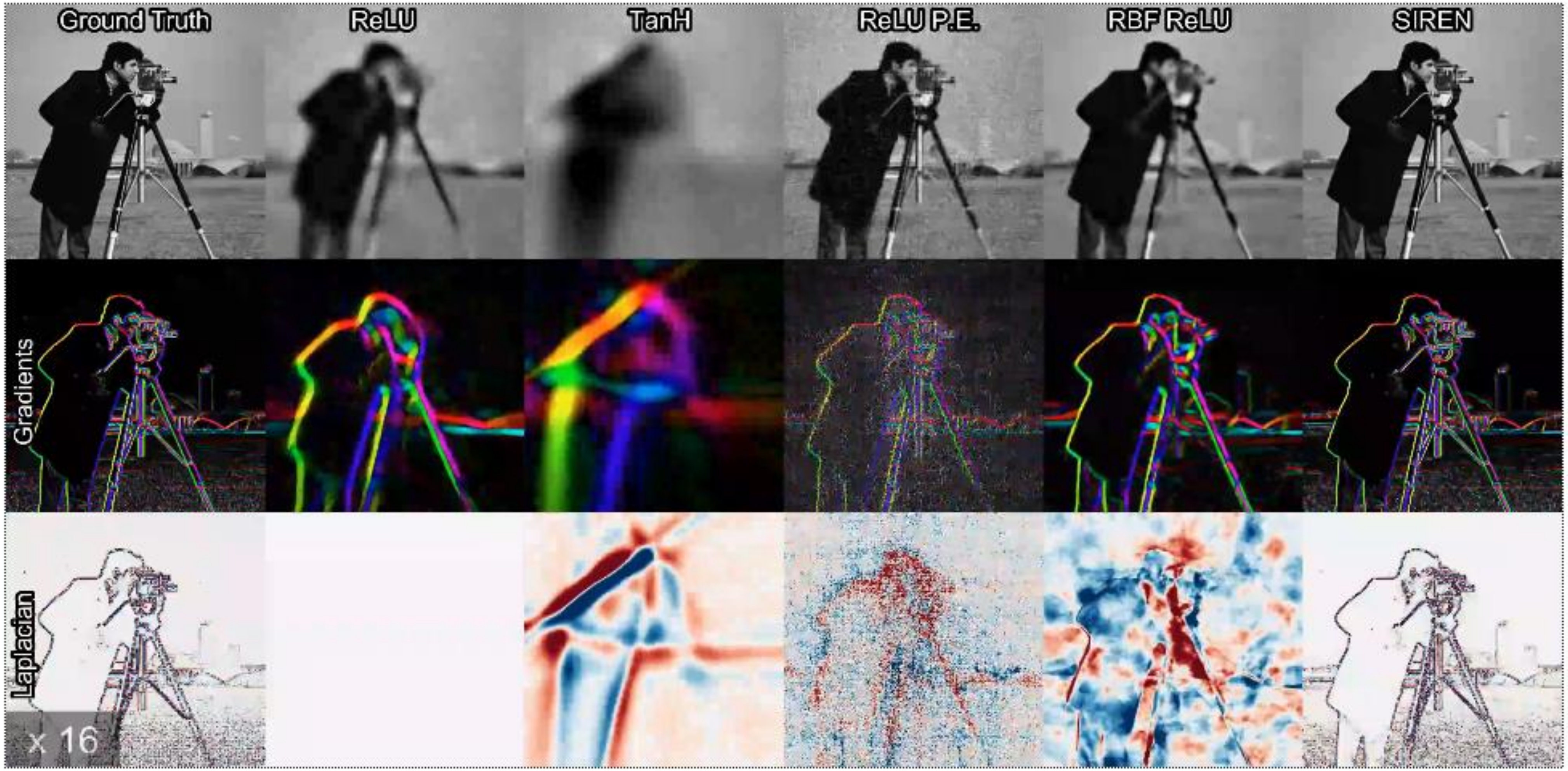
- Now, topics will be presented
- Topic assignment:
  - Non-binding poll to find most-wanted topics
  - Short discussion (to resolve conflicts)
  - Activate group choice in TUWEL -> first come, first serve
  - Switching is possible until the end of the week
  - Double assignment or groups if more students than topics

Deep Learning Fluid Simulation

Figure 7. Plume simulation with "Arch" geometry. *Left*: PCG. *Middle* small-model *Right*: this work.

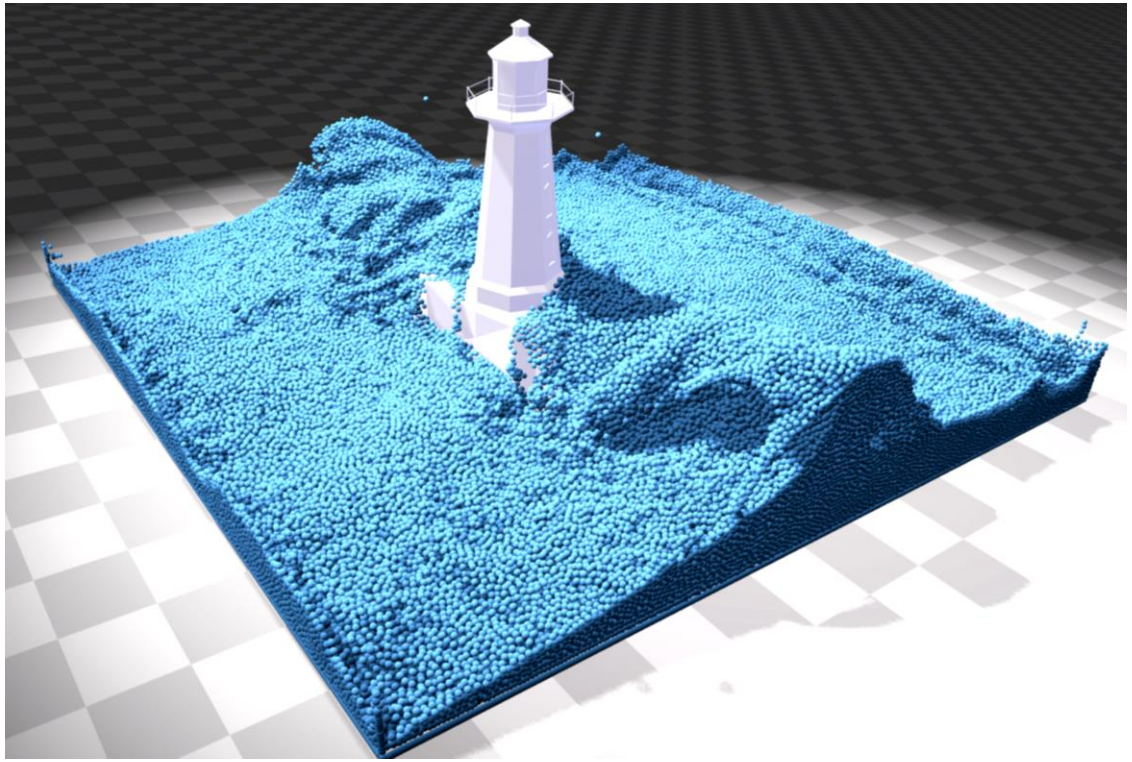
Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, Ken Perlin.
Accelerating Eulerian Fluid Simulation With Convolutional Networks

Neural Representation of 3D Data



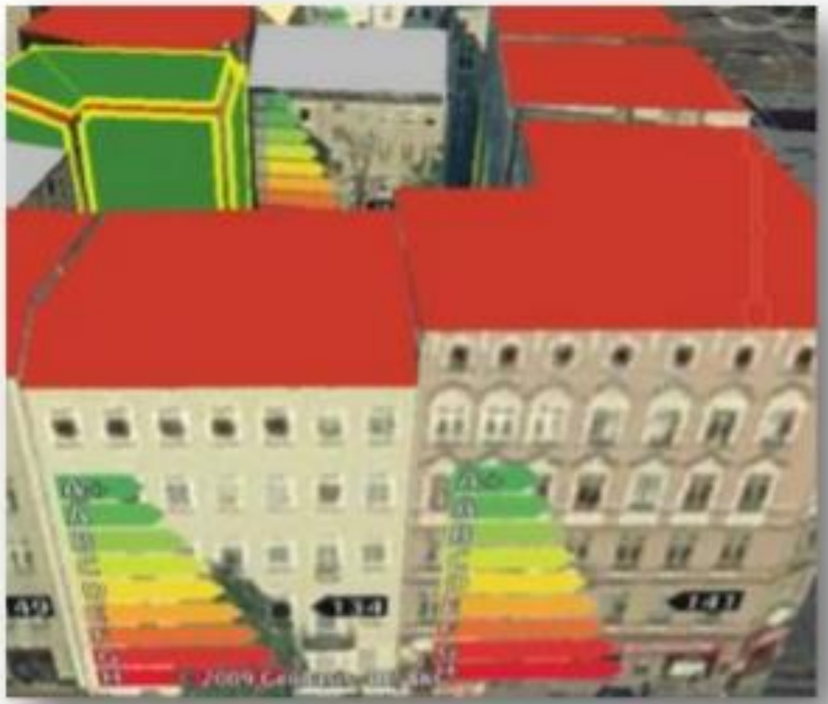
Vincent Sitzmann, Julien N. P. Martel, Alexander Bergman, David B. Lindell, Gordon Wetzstein.
Implicit Neural Representations with Periodic Activation Functions

- Many methods out there (SPH, Vortex, Position-based…)
- Often quite involved, material not very didactic
- Compare: which of them are
  - Robust (infrequent artifacts)
  - Versatile (handles walls, obstacles)
  - ***Not too hard to implement***
  - ***Implementations are explainable***

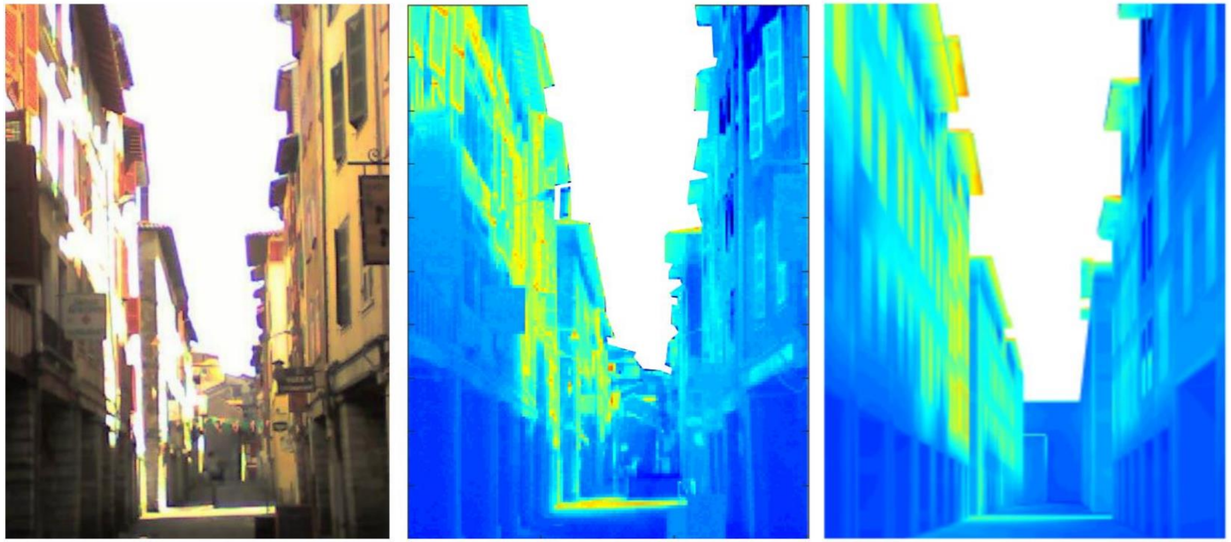


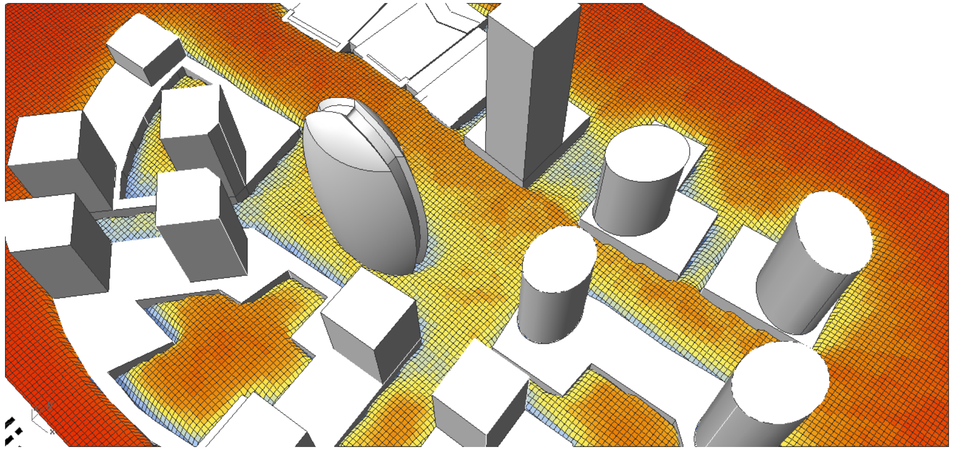
Semantic Enrichment of Urban Data

- 3D models available for cities, buildings…
- Look nice, but no semantic data! Needed for:
  - Simulating heat, noise dispersion
  - Planning, restoration…
- How to extract information on
  - Walls (thickness, shape)?
  - Windows (size, number)?
  - Material (insulation, absorption)?



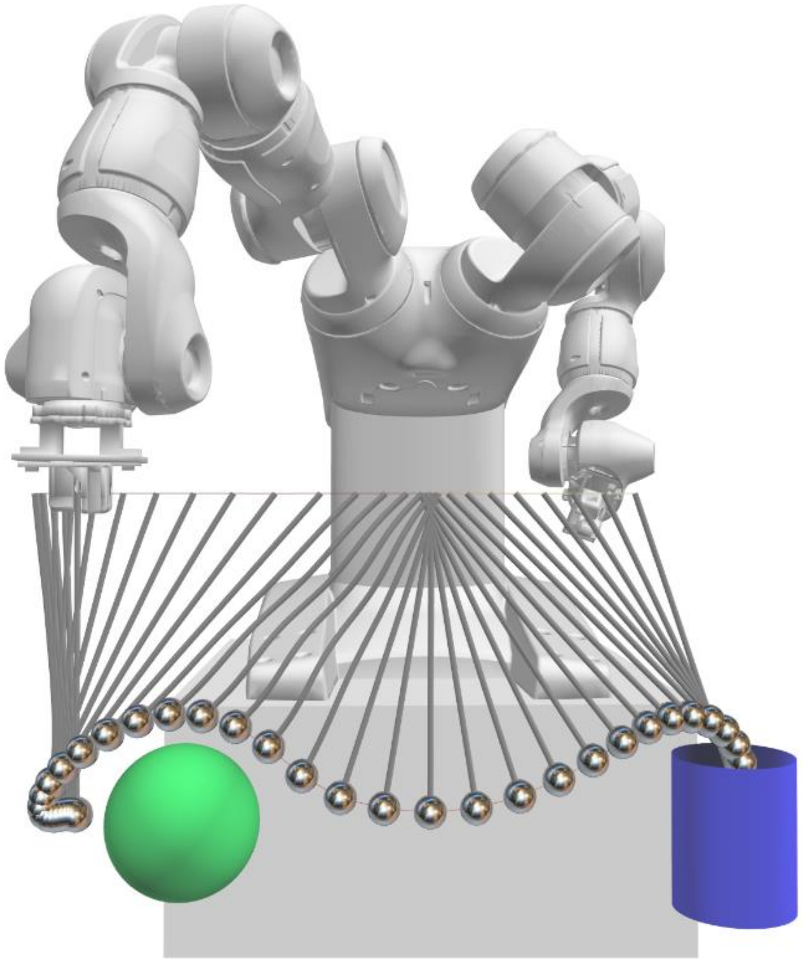
- Conduct a survey on recent advances in simulation and rendering of thermal radiation





Differentiable Simulation

$$\frac{\mathrm{d}\,\Phi(\mathbf{x}(\mathbf{q}))}{\mathrm{d}\,\mathbf{q}}$$

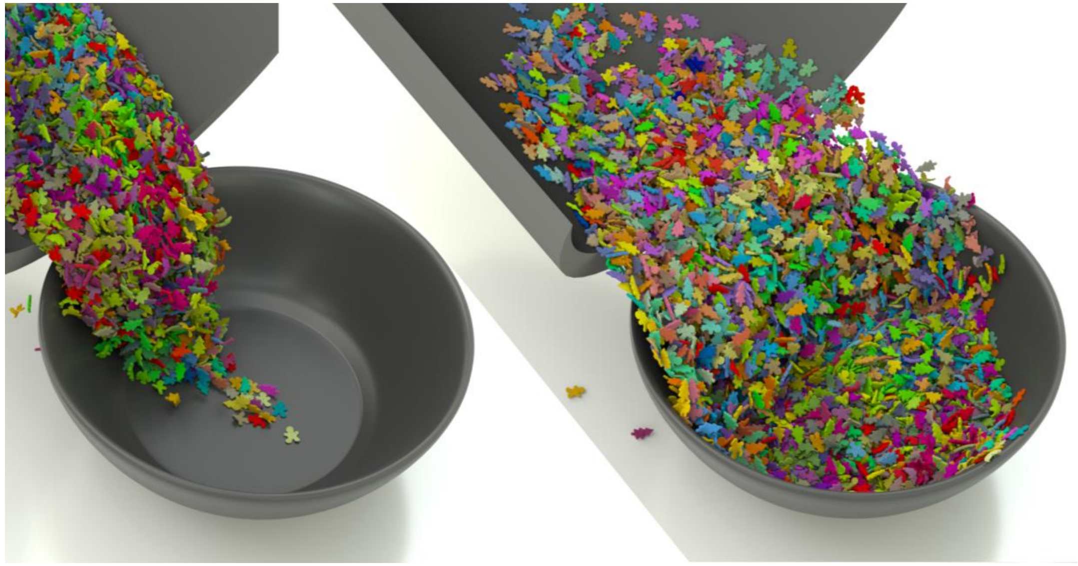


Liang et al., Differentiable Cloth Simulation for Inverse Problems. Advances in Neural Information Processing Systems 32 (2019)

Zimmermann et al., PuppetMaster: Robotic Animation of Marionettes. ACM Trans. Graph. 38, 4 (2019)

GPU-accelerated Simulation

developer.nvidia.com/flex



A Massively Parallel And Scalable Multi-GPU Material Point Method
Wang, et. al (SIGGRAPH 2020)

Implicit FEM and fluid coupling on GPU for interactive multiphysics simulation, Allard et al., SIGGRAPH 2011

- Provide an overview of offline (non-real-time) rendering techniques that take human perception into account.

(a) Error distribution

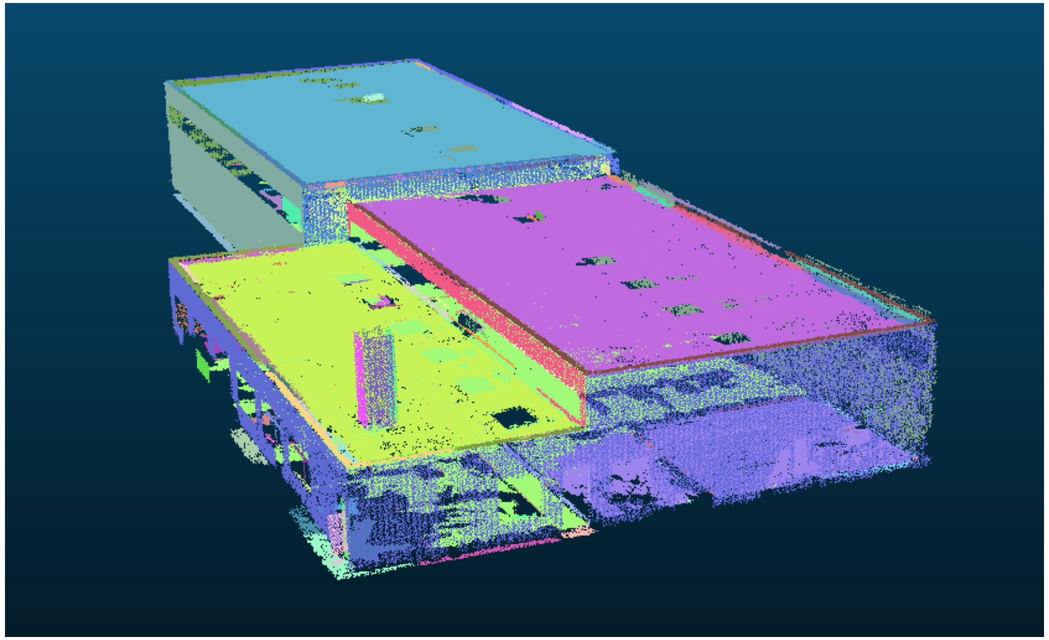
(b) After denoising

Rendering Astronomical Phenomena

- Provide an overview of techniques for rendering astronomical phenomena, such as black holes, nebulae, etc.

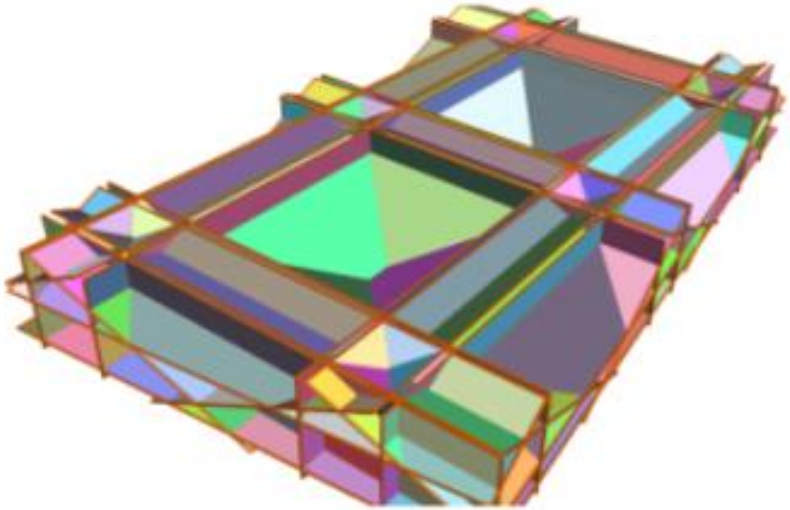
Planar segmentation of point clouds

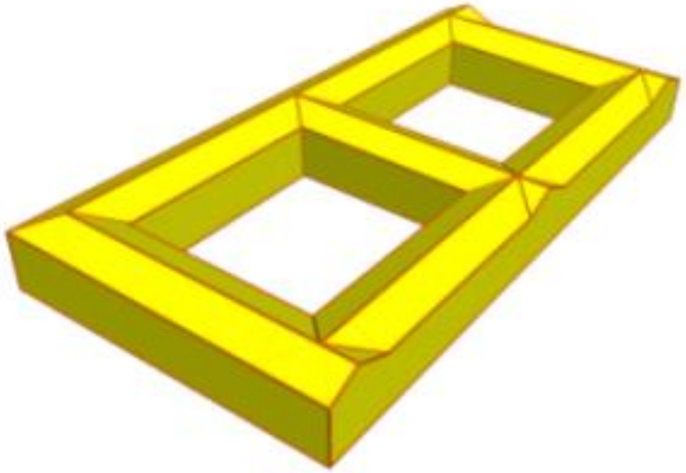
- Cluster point clouds into regions, where points are close to each other and lie nearly on the same plane
- E.g. Region Growing, RANSAC



* planar regions painted by unique color

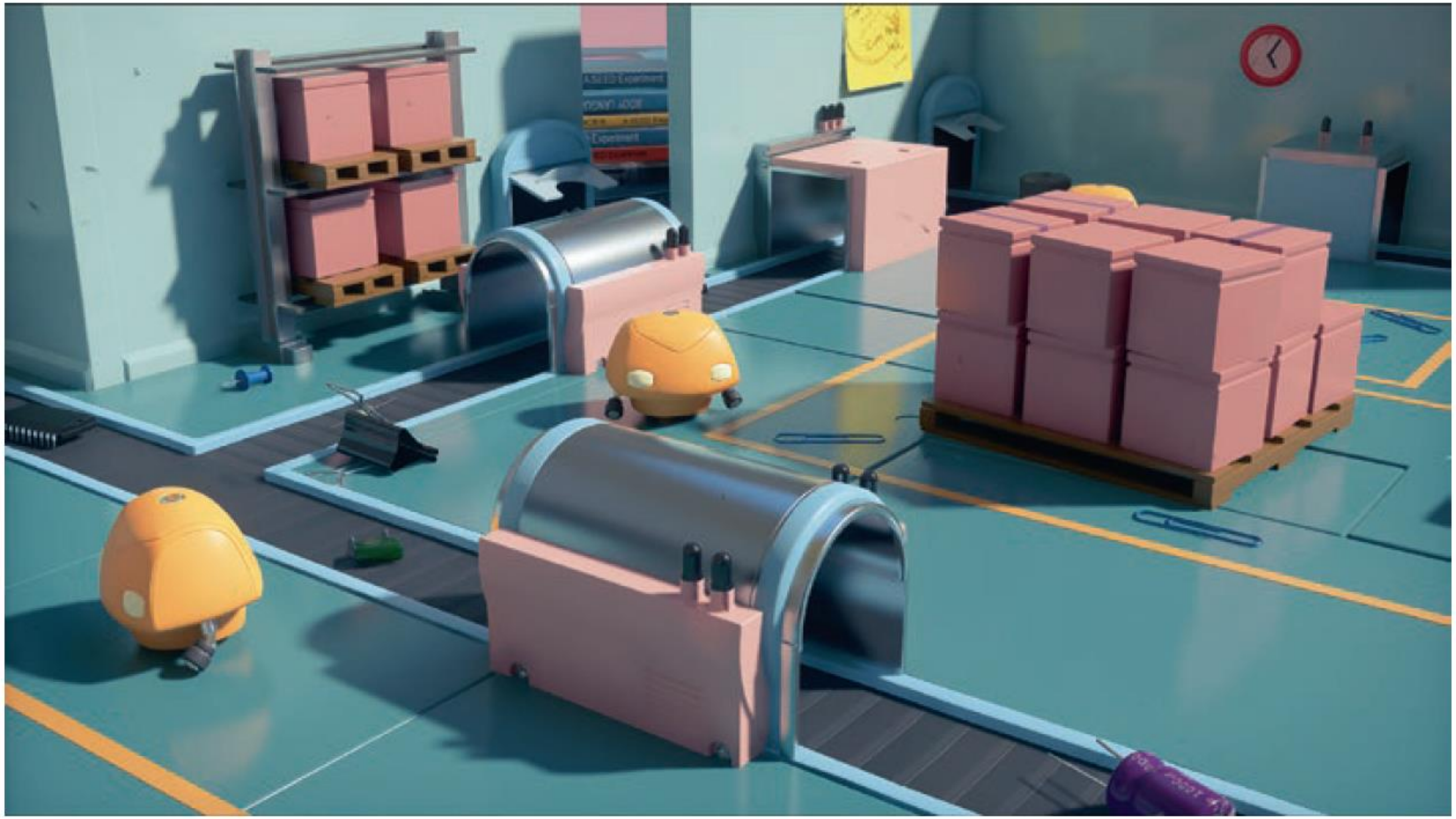
- Given a cellular complex, label cells or facets to extract a polygonal mesh





Nan, L., Wonka, P., 2017. “Polyfit: Polygonal surface reconstruction from point clouds”

Hybrid Rendering Techniques

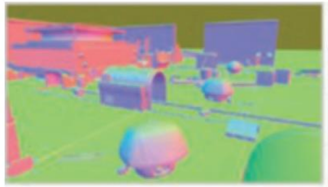



Direct Lighting
(Compute)


Reflections
(Ray Trace or Compute)


Transparency
& Translucency
(Ray Trace and Compute)


Post-Processing
(Compute)


G-Buffer
(Raster)

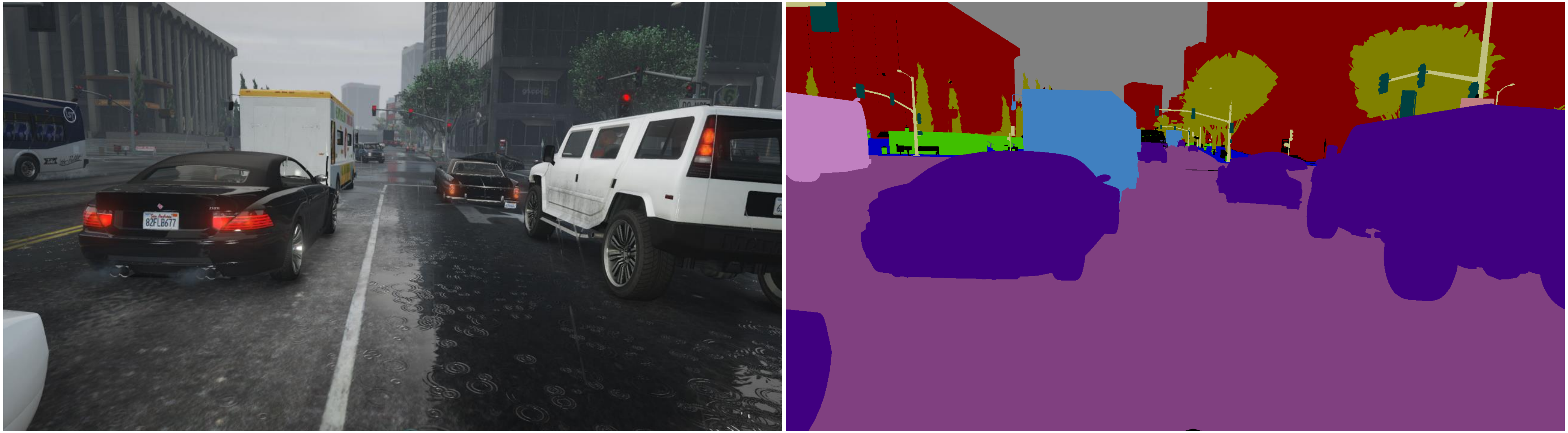

Direct Shadows
(Ray Trace or Raster)


Global Illumination
(Ray Trace and Compute)


Ambient Occlusion
(Ray trace or Compute)

Hybrid Rendering for Real-Time Ray Tracing
http://www.realtimerendering.com/raytracinggems/unofficial_RayTracingGems_v1.7.pdf

Deep Learning for Image Segmentation



Playing for Data: Ground Truth from Computer Games
https://download.visinf.tu-darmstadt.de/data/from_games/

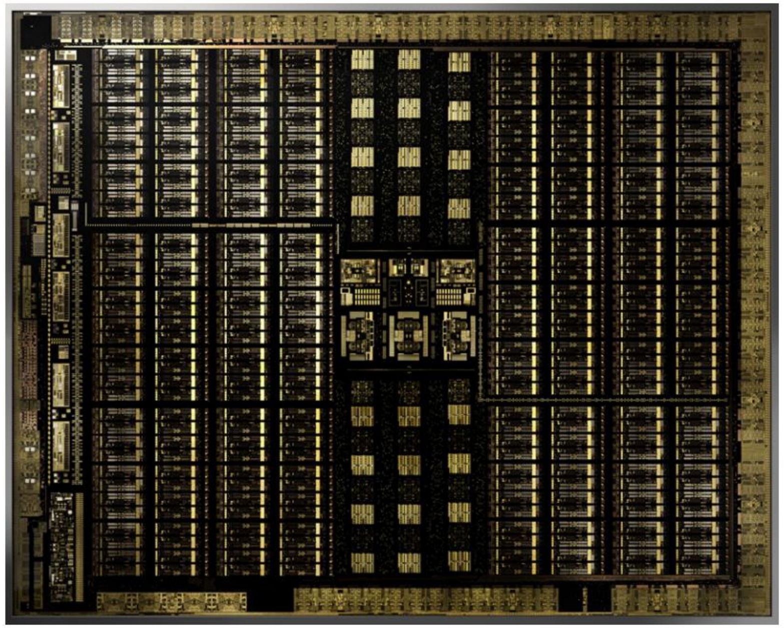
Frame-Analysis of High-End Game Graphics + Techniques

- First, pick **3 to 5 games** with **high-end 3D graphics** and analyze how a frame is rendered!
  - Games should not be older than 5 years
  - See: Adrian Courrèges’ Graphics Studies Compilation
  - Describe common patterns or approaches between different games/game engines.
- Furthermore, extract **common techniques** used among these games/game engines!
  - Describe the most common techniques in detail!
  - Try to argue why these techniques are useful across different games/game engines.
  - You still need ≥10 scientific papers besides the (probably web-based) frame-analysis articles!

Hardware Units of GPUs

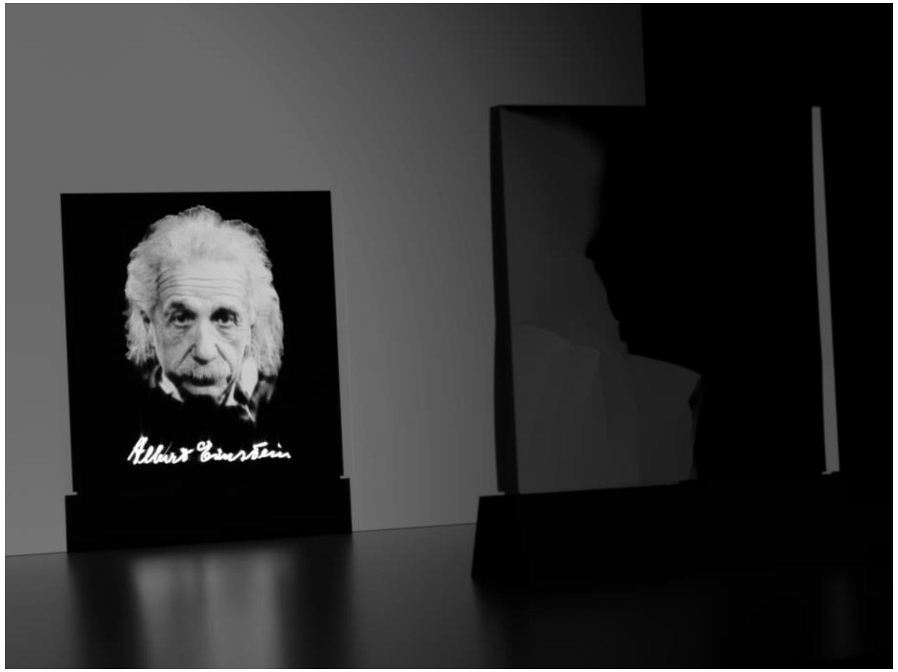
- Investigate the hardware units of GPUs and which operations they accelerate. Also analyze the different levels of memory and cache.
  - Texture Units, Render Output Units, Warp Scheduler, …
  - L1 Cache, L2 Cache, Instruction Cache, Registers, …
  - Other specialized cores/units (e.g. RTX cores, …)
  - Focus on **modern** GPUs
    - Which of these units are implemented in hardware (i.e. hardware-accelerated)
    - Which operations to these units accelerate in hardware in particular?
    - Why is hardware-acceleration required for these operations?

NVIDIA Turing TU102 GPU

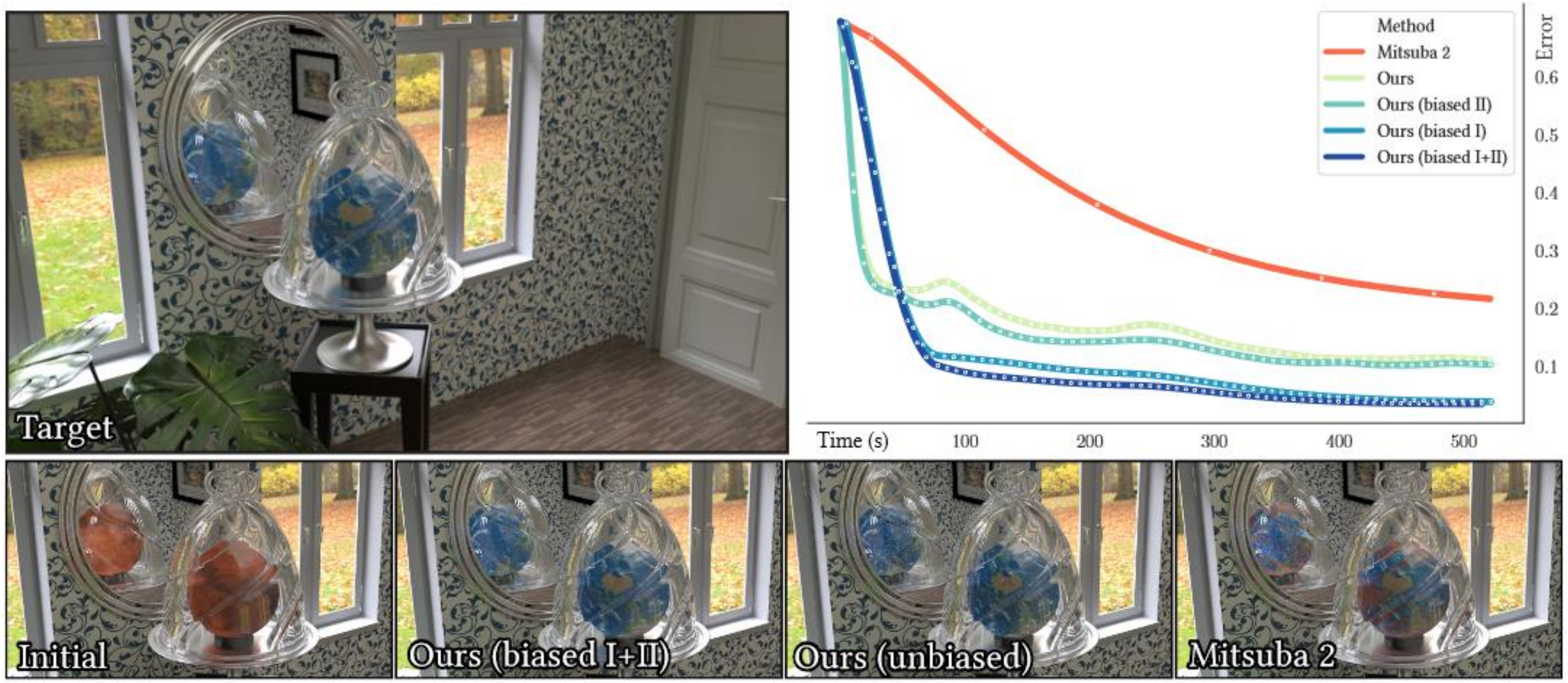


Inverse / Differentiable Rendering

- Use cases for inverse / differentiable rendering
- Discuss ways of calculating an objective function which incorporates scene parameters



Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-contrast computational caustic design.



Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering.

Alternative Machine Learning Technologies

- Discuss pros/cons of alternatives to Python/CUDA for ML
  - TensorFlow for Swift (archived)
  - Vulkan backend for PyTorch (prototype)
  - AMD ROCm backend for PyTorch (beta)
  - mlpack (C++ and CPU parallelization)

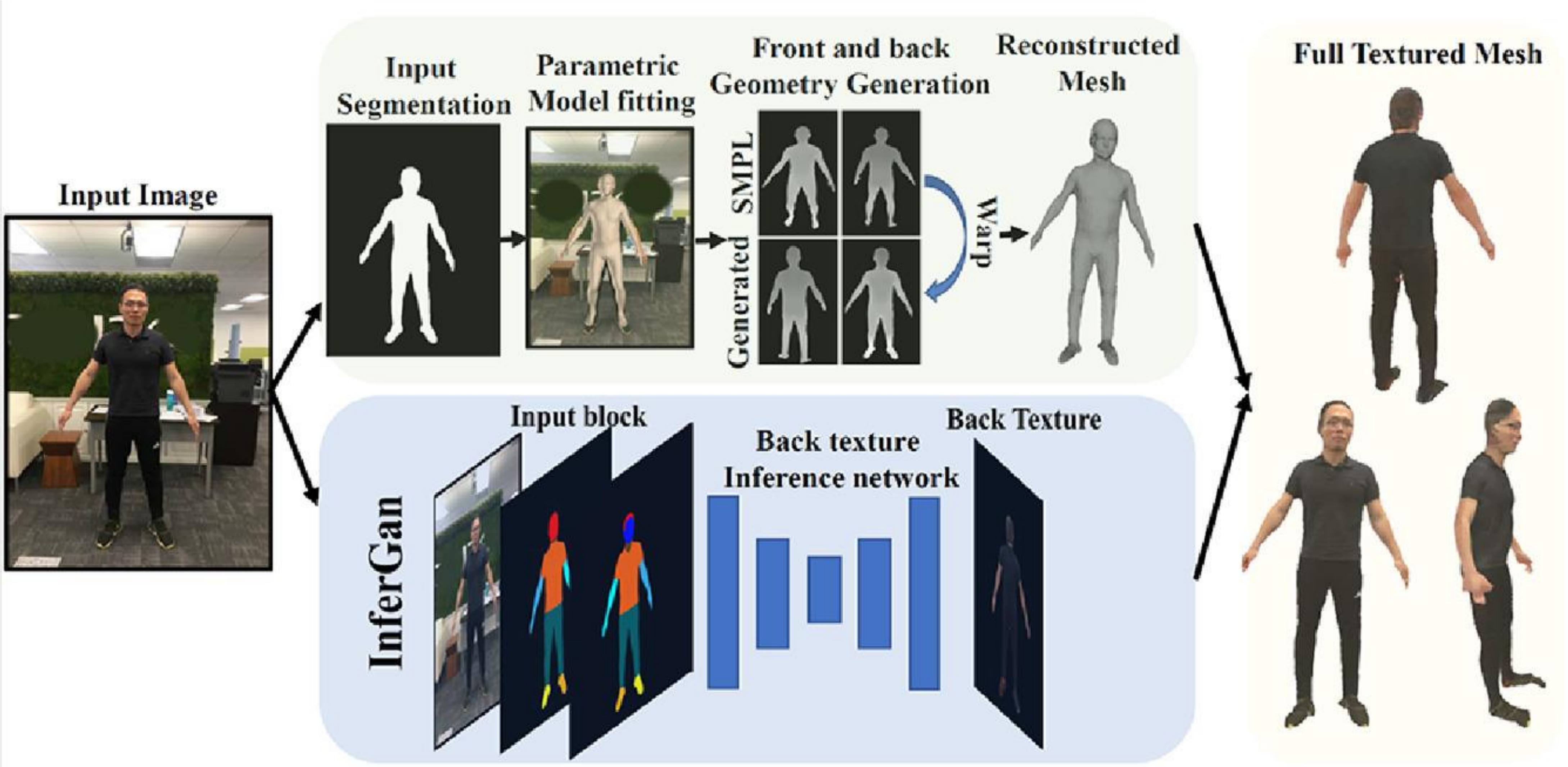
https://blog.tensorflow.org/2018/04/introducing-swift-for-tensorflow.html

mlpack

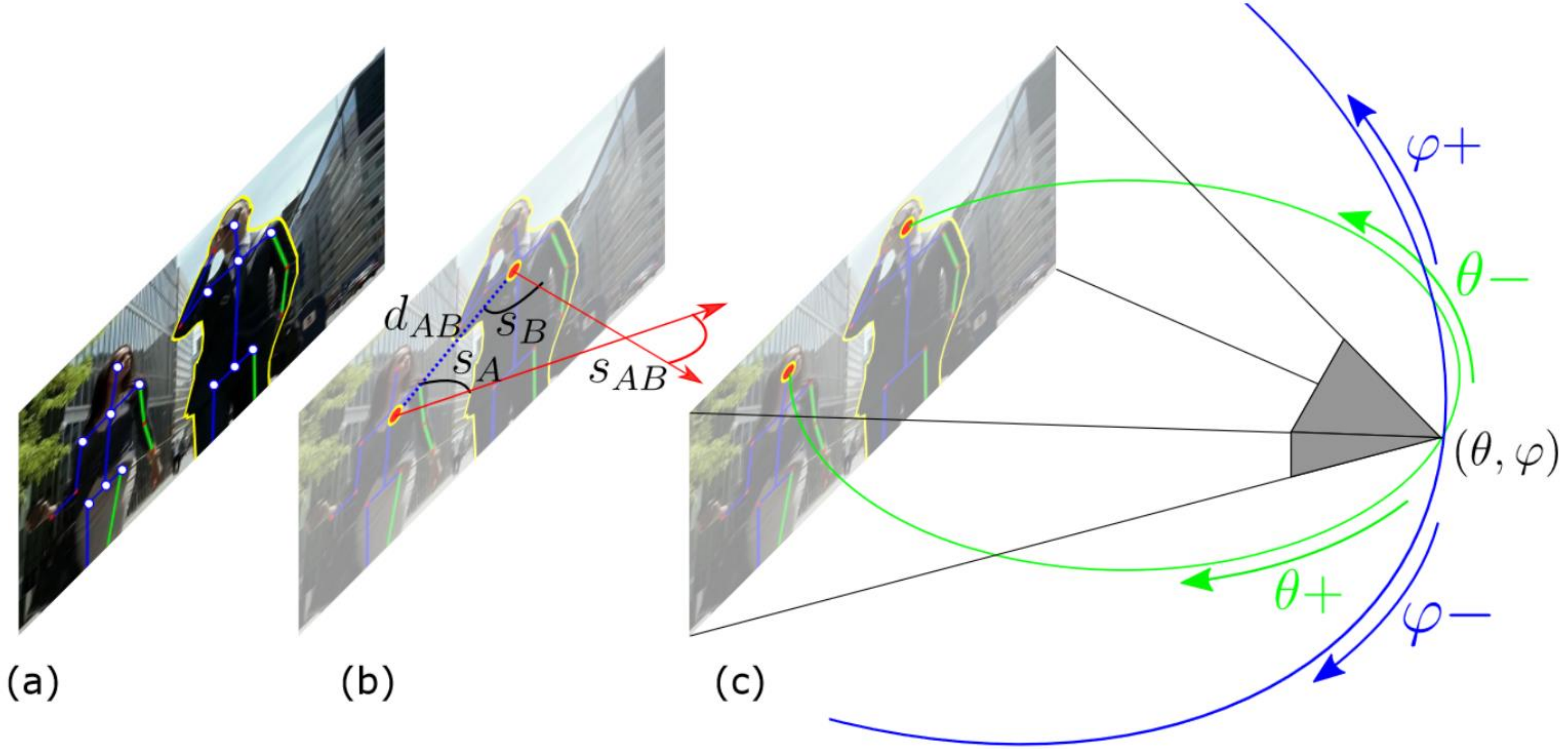
Designing and building the mlpack open-source machine learning library
https://arxiv.org/abs/1708.05279

https://pytorch.org/tutorials/prototype/vulkan_workflow.html

3D Human Shape Estimation

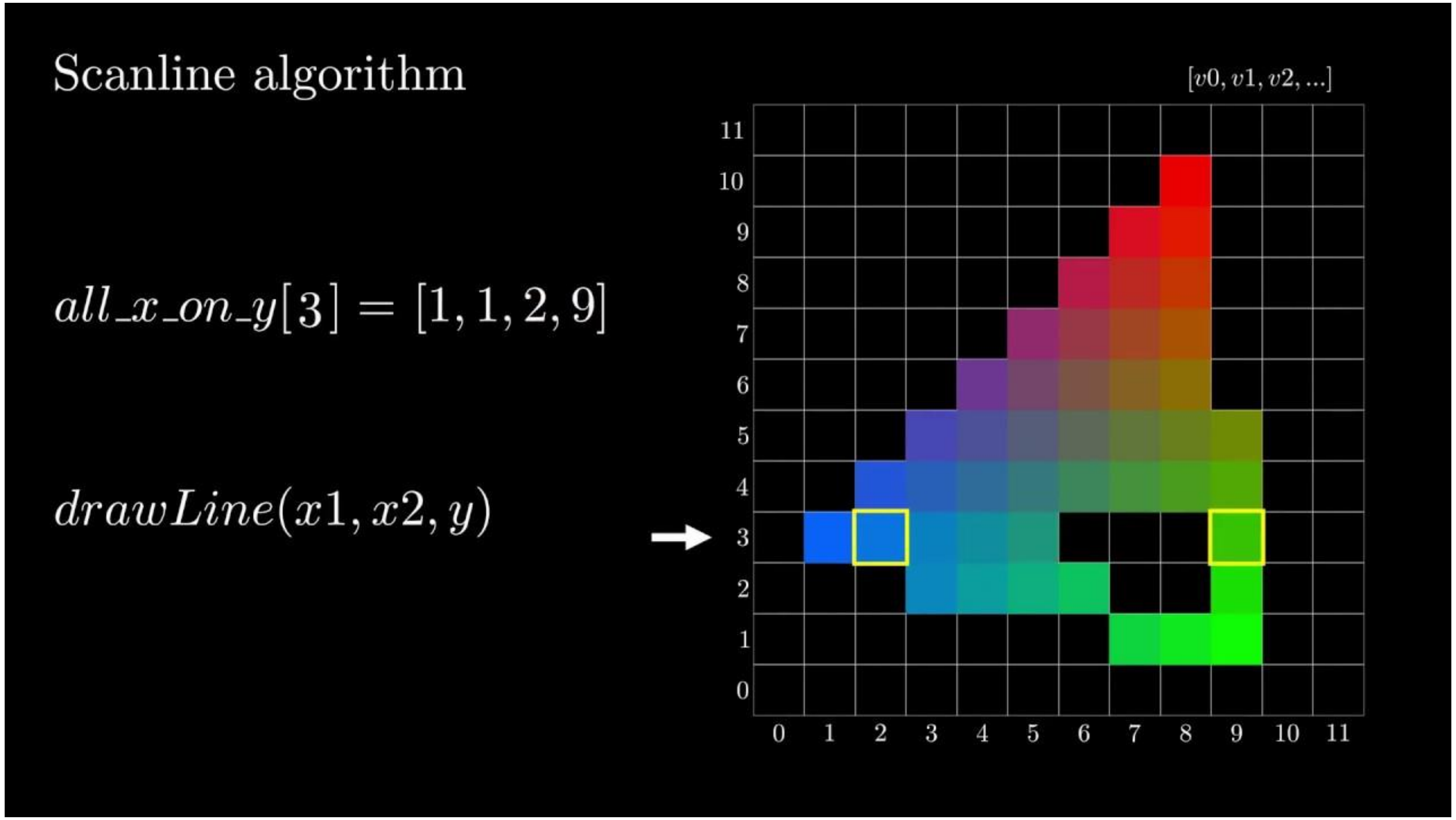


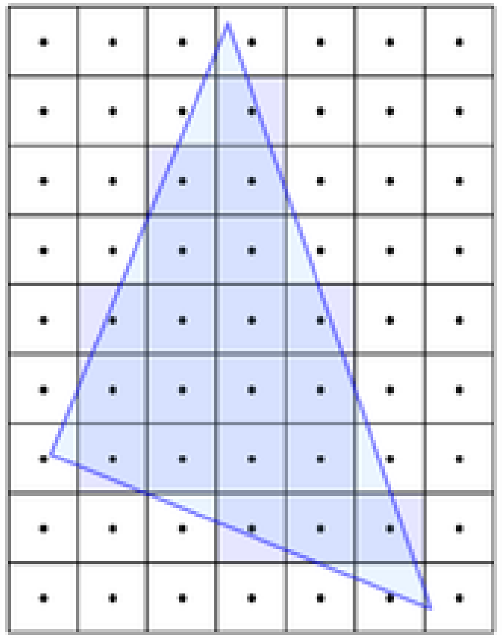
Li Z, Chen L, Liu C, Zhang F, Li Z, Gao Y, Ha Y, Xu C, Quan S, Xu Y. Animated 3D human avatars from a single image with GAN-based texture inference. Computers & Graphics. 2021.



Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. 2020. Example-driven virtual cinematography by learning camera behaviors. ACM Trans. Graph. 39, 4, Article 45 (July 2020), 14 pages.

- Rendering triangles, lines, points without using GPU primitives or ray tracing.



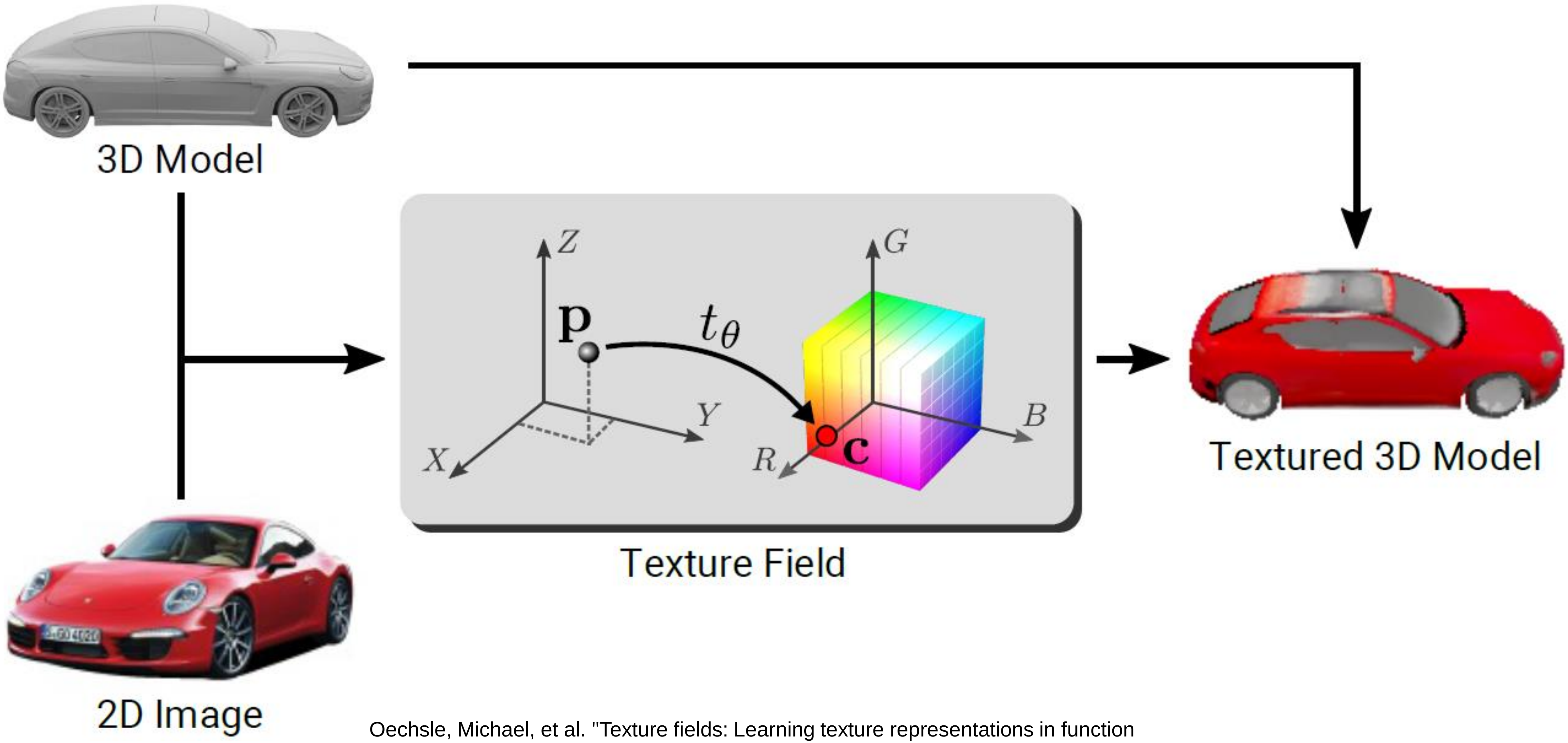


Cache-Friendly Programming

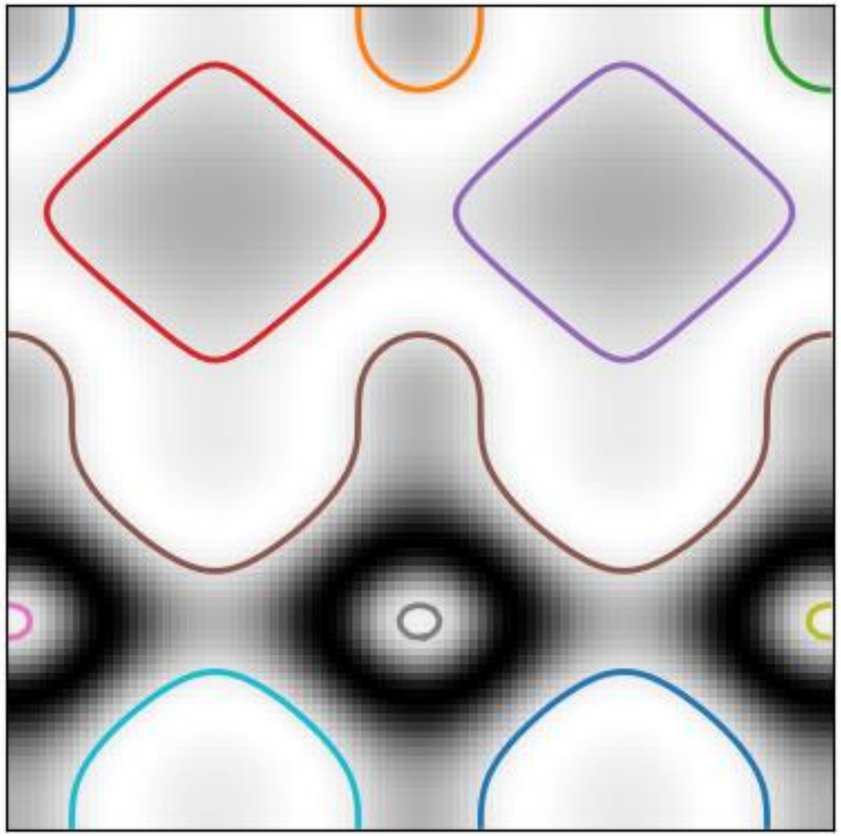
- How to optimize based on cache behavior?
- Research cache-friendly algorithms with respect to graphics

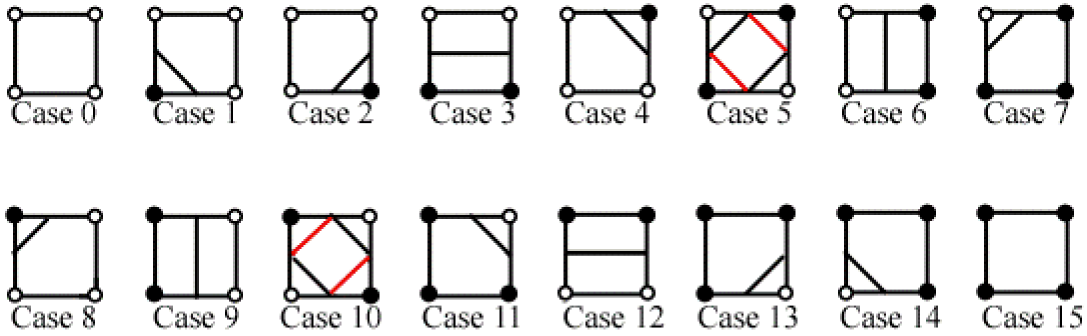
	1	2	3	4
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12
4	4	8	12	16

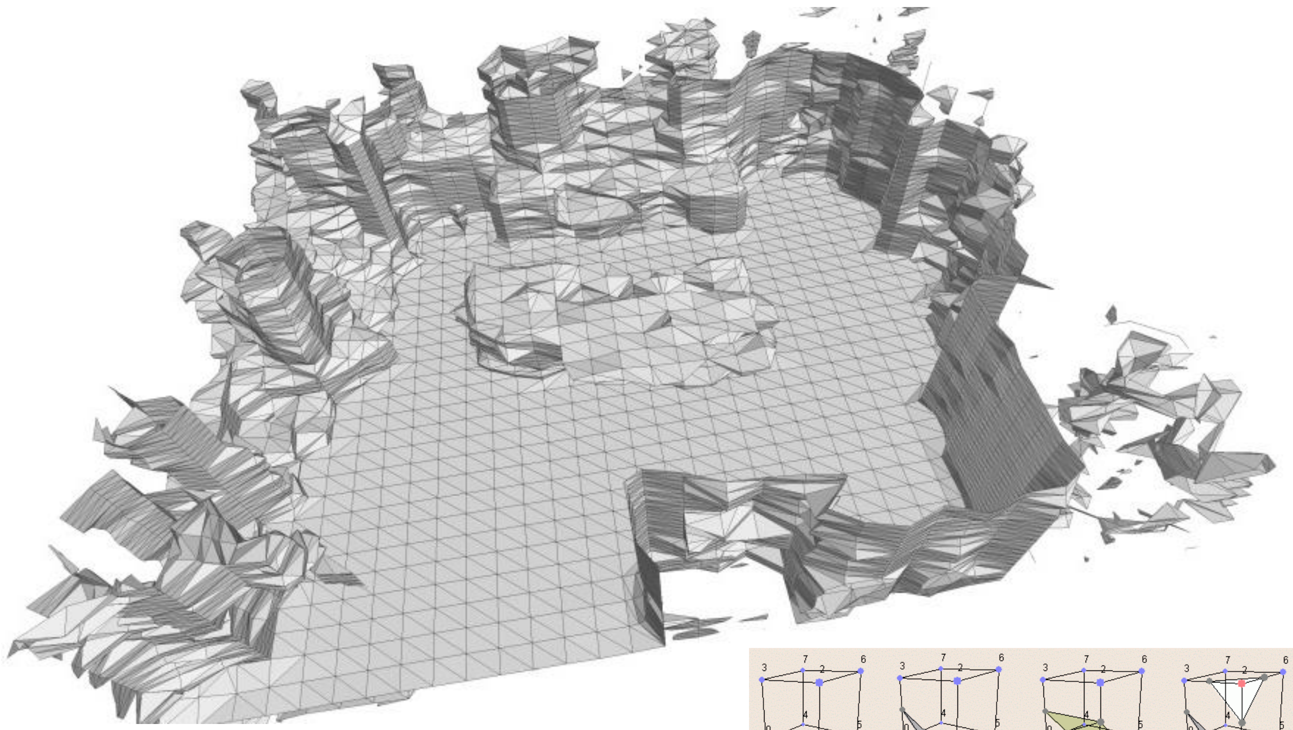
Colored Reconstruction

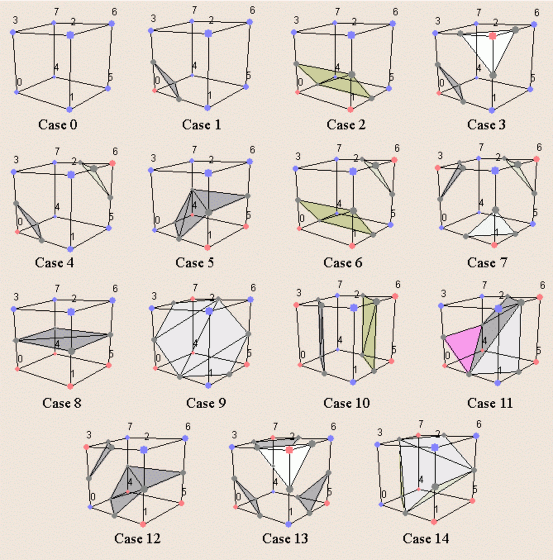


Oechsle, Michael, et al. "Texture fields: Learning texture representations in function space." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.




Case 0
Case 1
Case 2
Case 3
Case 4
Case 5
Case 6
Case 7
Case 8
Case 9
Case 10
Case 11
Case 12
Case 13
Case 14
Case 15




Case 0
Case 1
Case 2
Case 3
Case 4
Case 5
Case 6
Case 7
Case 8
Case 9
Case 10
Case 11
Case 12
Case 13
Case 14

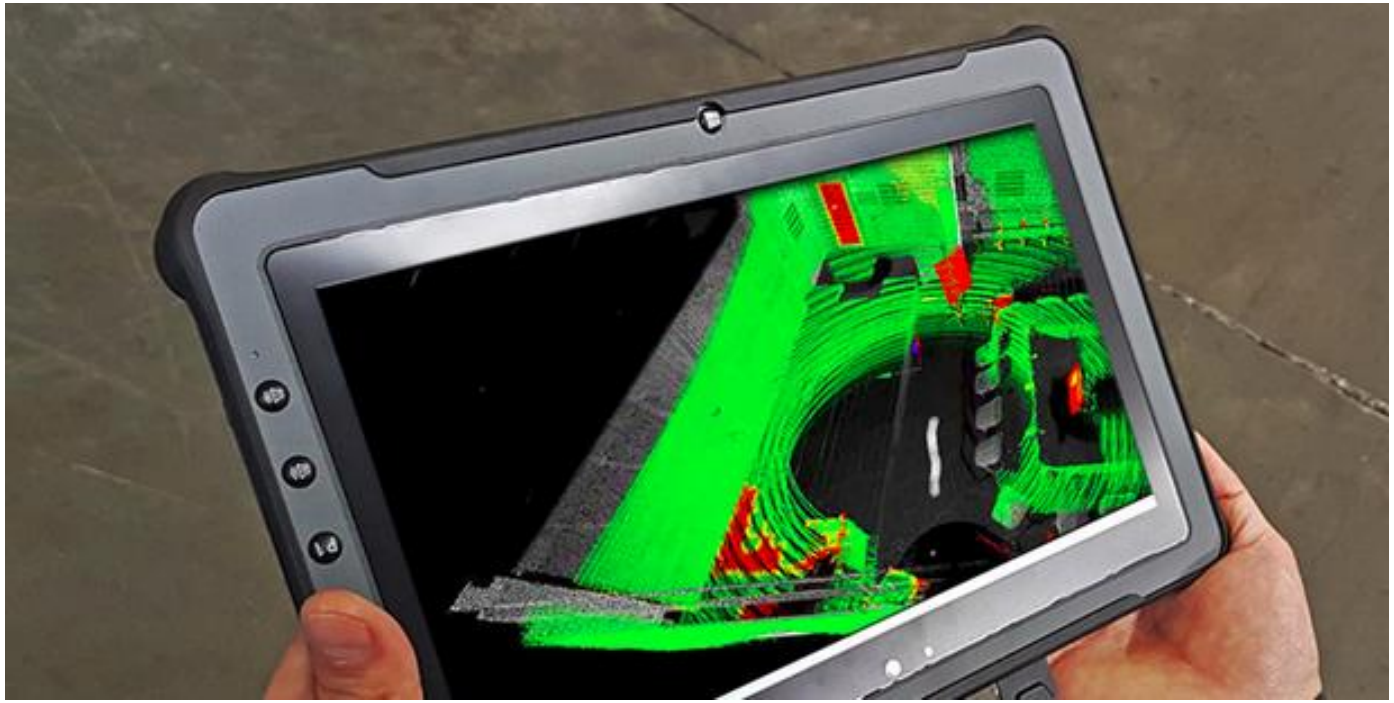
Classify Objects in Point Clouds

- Machine learning algorithms for 3D scanned data
- Detect partial objects and their pose (location+orientation in 3D)

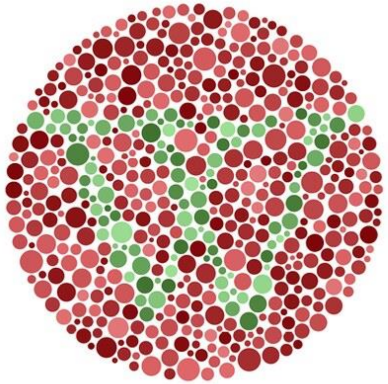
Real-time Change Detection

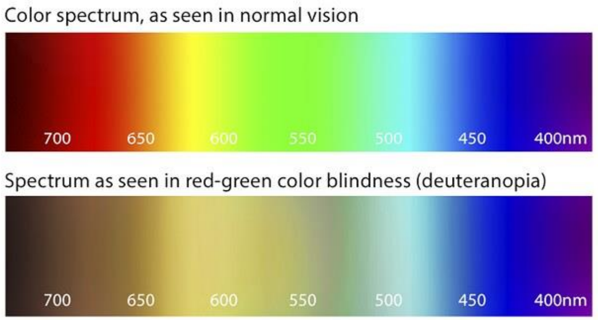
- Mobile App shows AR changes to scanned 3D model in real-time

- Requirements:
- 3D Occupancy Maps
- Sensor noise tolerance
- Clustering segments
- Real-time performance

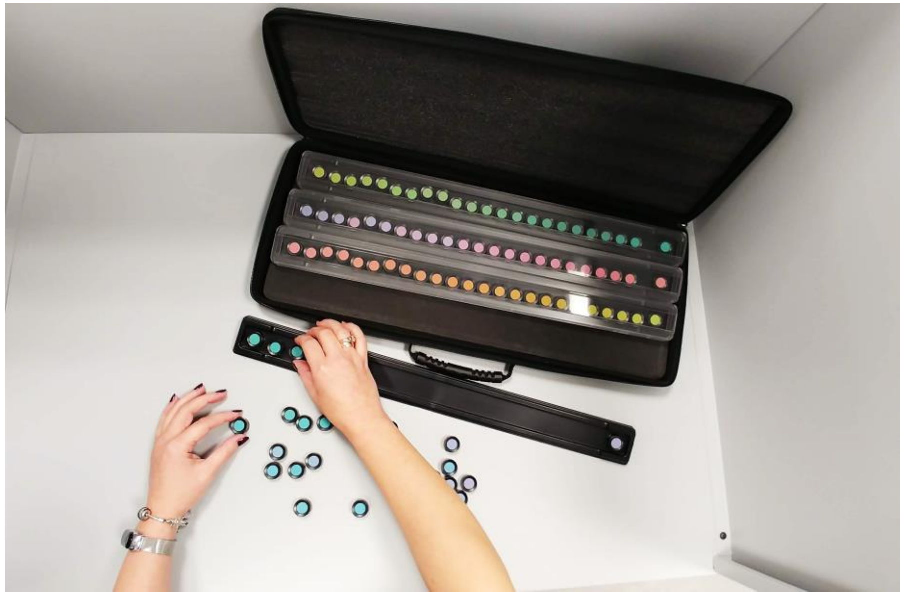


- Principles of color perception
- Computerized/**gamified** color vision tests
- Color vision Tests **for young kids**





https://www.flickr.com/photos/webstyleguide/25379791138



https://commons.wikimedia.org/wiki/File:Farnsworth%E2%80%93Munsell_Hue_Color_Vision_Test,_Material_and_Finishing_Laboratory.jpg

Good starting point: Hasrod, Nabeela, and Alan Rubin. "Colour vision: A review of the Cambridge Colour Test and other colour testing methods." *African Vision and Eye Health* 74.1 (2015): 7.

Topic Assignment

1. Deep Learning Fluid Simulation
2. Neural Representation of 3D Data
3. Simplified Fluid Simulations for Real-Time Rendering
4. Semantic Enrichment of Urban Data
5. Procedural Plant Generation
6. Simulation and Rendering of Thermal Radiation
7. Differentiable Simulation
8. GPU-accelerated Simulation
9. Perceptual Rendering Techniques
10. Rendering Astronomical Phenomena
11. Planar segmentation of point clouds
12. Labeling problem in polygonal reconstruction
13. Hybrid Rendering Techniques
14. Deep Learning for Image Segmentation
15. Frame-Analysis of High-End Game Graphics + Techniques
16. Hardware Units of GPUs
17. Inverse / Differentiable Rendering
18. Alternative Machine Learning Technologies
19. 3D Human Shape Estimation
20. 3D Pose Reconstruction
21. Software Rasterization
22. Cache-Friendly Programming
23. Colored Reconstruction
24. Iso-Surface Extraction
25. Classify Objects in Point Clouds
26. Real-time Change Detection
27. Color Vision Tests for Kids

- Non-binding poll to show most-wanted topics
- Short discussion
- Activate group choice in TUWEL -> first come, first serve

Next Steps

- Contact your supervisor ASAP (name in left lower corner of slide)
- Find mail addresses here: https://www.cg.tuwien.ac.at/staff/
- Discuss literature list with your supervisor
- Submit the literature list in TUWEL

- Questions?