Final Project List

Scoring criteria

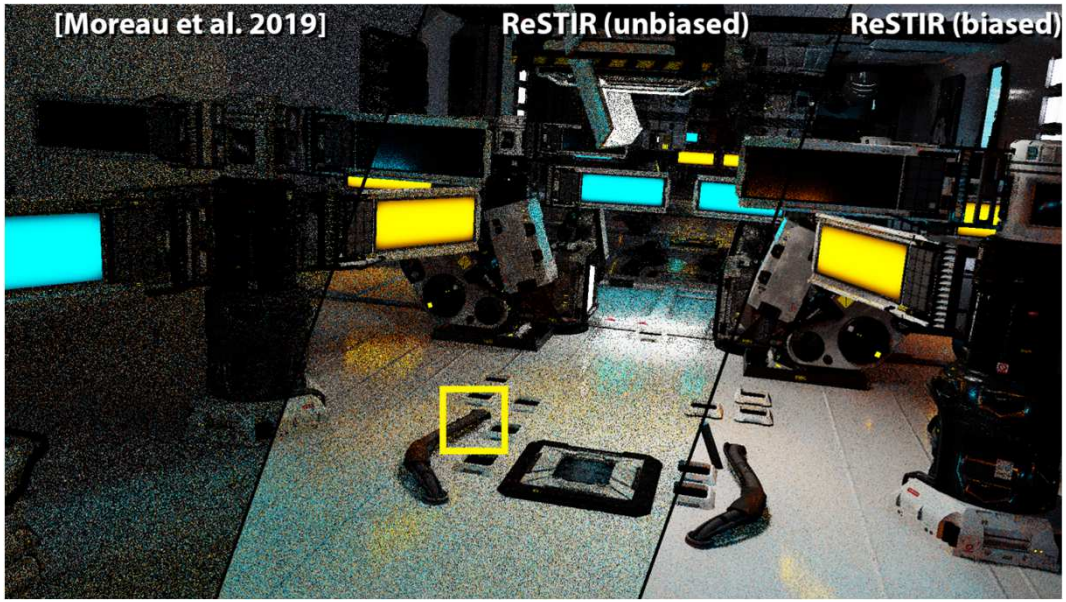
- The projects are categorized into A+, A, B and C levels.
- If you choose to accomplish level C, the maximum score is 80% of the full score.
- If you choose to accomplish level B, the maximum score is 100% of the full score.
- If you choose to accomplish level A, the maximum score is 120% of the full score.
- If you choose to accomplish level A+, the maximum score is 140% of the full score.
- We will score each individual according to their percentage of contributions

Real-Time Fluid Simulation (A+)

- **Simulate smoke by a more accurate advection solver**
  - Implement the stable fluids solver
  - Modify the advection by the reflection solver
  - Implement them on GPU using CUDA
  - GPU-based volume renderer of velocity field
- **Reference**
  - https://dl.acm.org/doi/pdf/10.1145/311535.311548
  - https://jzehnder.me/publications/advectionReflection/

Reservoir Spatio-Temporal Importance Resampling (A+)

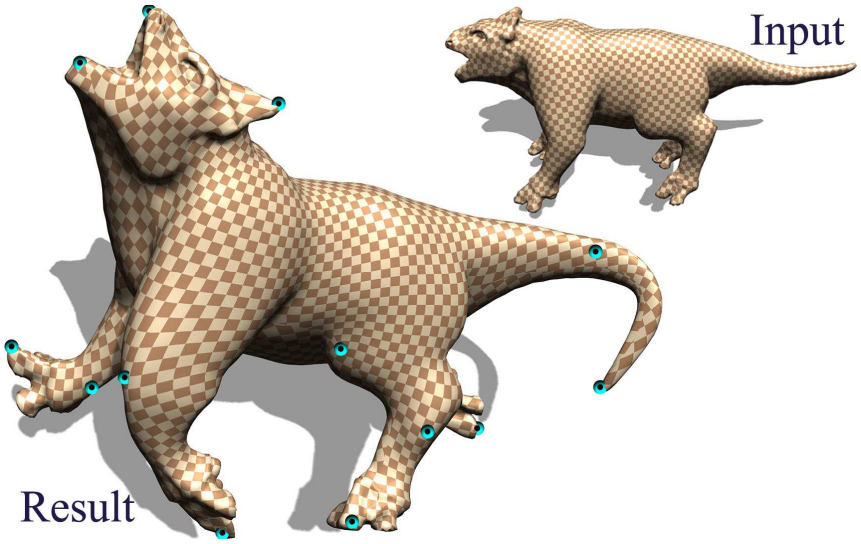
- **Rendering surface with ReSTIR efficiently**
  - Efficient sample many lights.
  - Reuse samples from other pixels.
  - Probably cross-sample instead of temporal.
- **Reference**
  - [2020] Spatiotemporal reservoir resampling for real-time ray tracing
  - [2021] ReSTIR GI Path Resampling for Real-Time Path Tracing



Real-time Locally Injective Volumetric Deformation(A+)

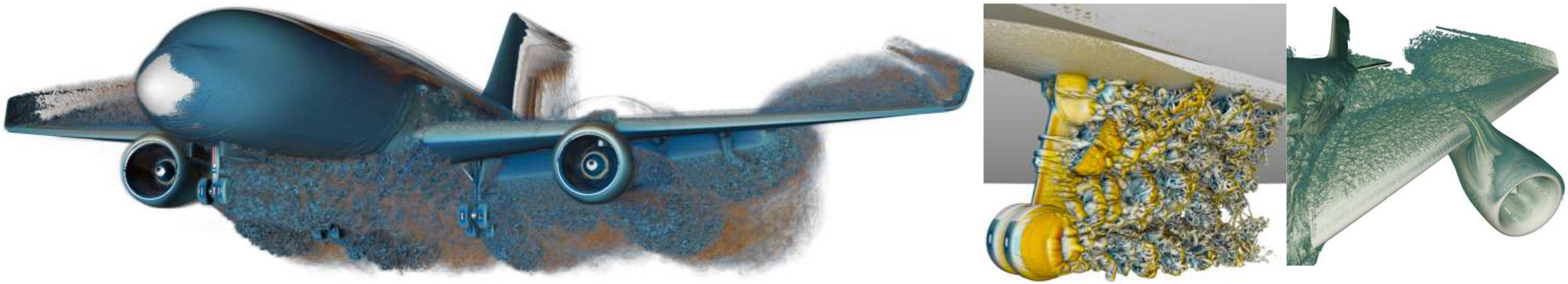
- **Interactive volumetric meshless shape deformation**
  - Locally smooth
  - Parallel GPU (CUDA) implementation

- **Reference**
  - [2021] Real-time Locally Injective Volumetric Deformation



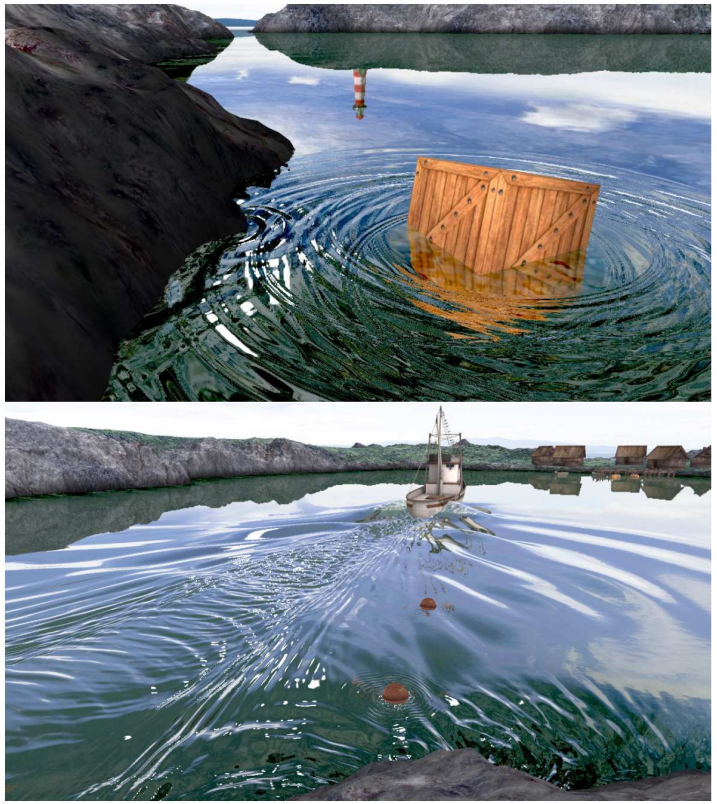
Multi-Resolution Isosurface Rendering (A)

- **Rendering isosurface with ray tracing based techniques**
  - Handle multi-level discrete volume data
  - Design transfer functions for isosurface rendering
  - Pre-integrated transfer function
- **Reference**
  - [2021] Ray Tracing Structured AMR Data Using ExaBricks



Advanced Water Wave (A)

- **Advanced water wave animation**
  - Implement advanced wave particles or procedural methods to solve the wave equation
  - Render the water surface for example with OpenGL and texture
  - Real-time required
- **Reference**
  - [2017] Water Wave Packets
  - [2018] Water Surface Wavelets

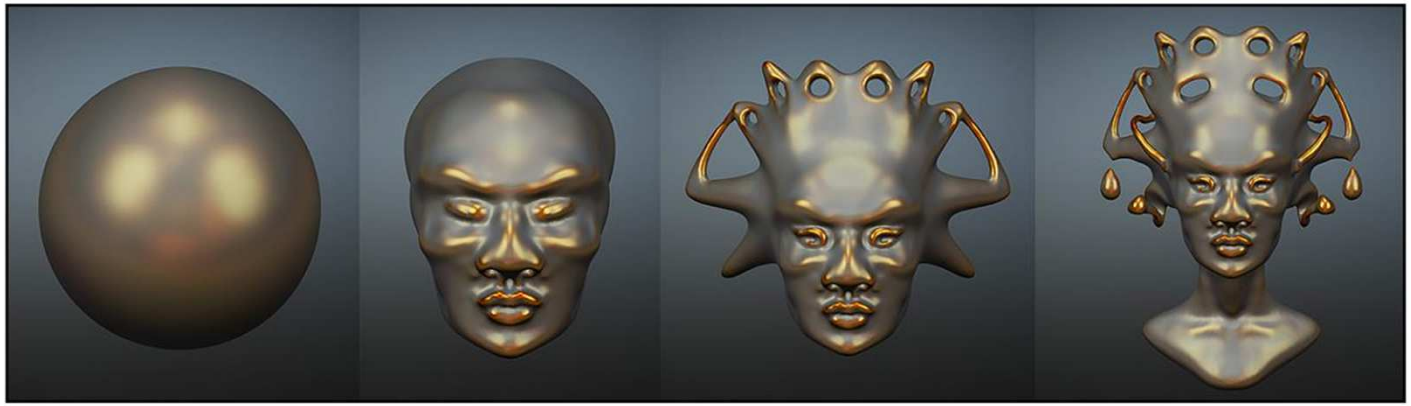


Interactive Sculpting (A)

- Interactive sculpture visual tool
- Dynamic changes of mesh structure

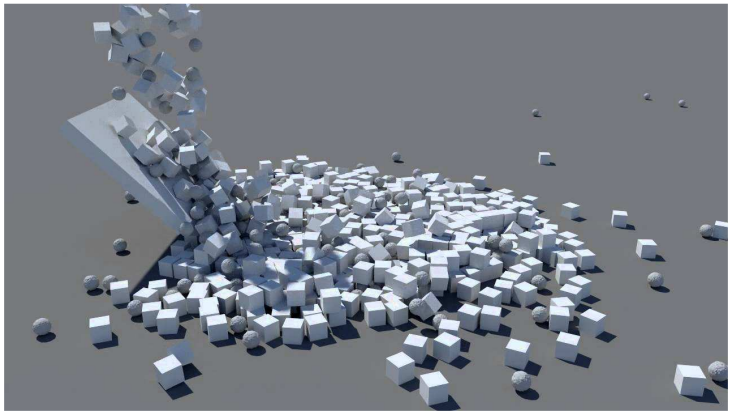
Reference:

- Stanculescu L, Chaine R, Cani M P. Freestyle: Sculpting meshes with self-adaptive topology[J]



Massive Rigid-Body Simulation (A)

- **Implement the massive rigid body simulation**
  - Basic transformation: translation + rotation over time
  - Collision detection
  - Handling colliding and resting contacts
- **Reference**:
  - Iterative Dynamics with Temporal Coherence
  - Game Physics in One Weekend

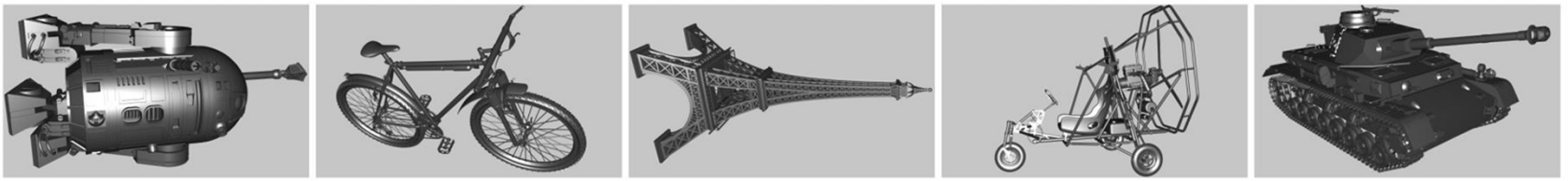


Deformable Solids Simulation (A)

- **Deformable body simulation**
  - Implement a deformable body simulation with collision
  - Multiple energy models and their comparisons
  - (Preferably) Use implicit euler for time integration.
- **Reference**:
  - https://graphics.pixar.com/library/DynamicDeformablesSiggraph2020/paper.pdf
  - https://viterbi-web.usc.edu/~jbarbic/femdefo/barbic-courseNotes-modelReduction.pdf

Ray Tracing NURBS Surface (A)

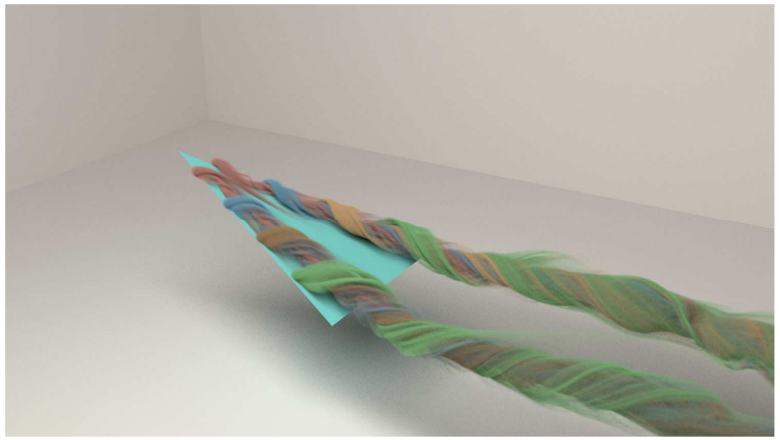
- **Rendering continues NURBS surface directly**
  - No meshing is required for rendering
  - Implement ray-surface intersection algorithm
  - Integrate into the ray-tracing framework with global illumination



- **Reference**
  - https://www.mattkeeter.com/projects/mrep/

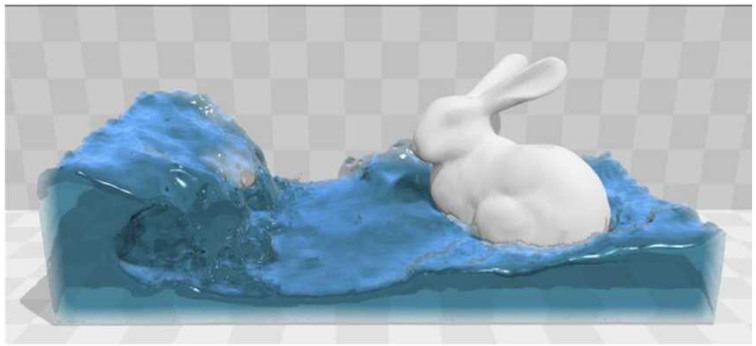
Realistic Smoke Rendering (A)

- **Rendering smoke by volumetric techniques**
  - Volume data saved in VDB files
  - Multiple scattering
  - Shadow effects
  - Combined with surface rendering
- **Reference**
  - Paper: [2017] Spectral and Decomposition Tracking for Rendering Heterogeneous
  - Paper: [2018] Monte Carlo methods for volumetric light transport simulation
  - Library: OpenVDB, NanoVDB.



Liquid Simulation by SPH Method (B)

- **Implement a 3D incompressible SPH solver for particle-based liquid simulation**
  - A fixed-domain neighbor search engine.
  - Parallelized Position Based Fluids solver.
  - Surface extraction for rendering or render particles
- **Reference**
  - https://mmacklin.com/pbf_sig_preprint.pdf

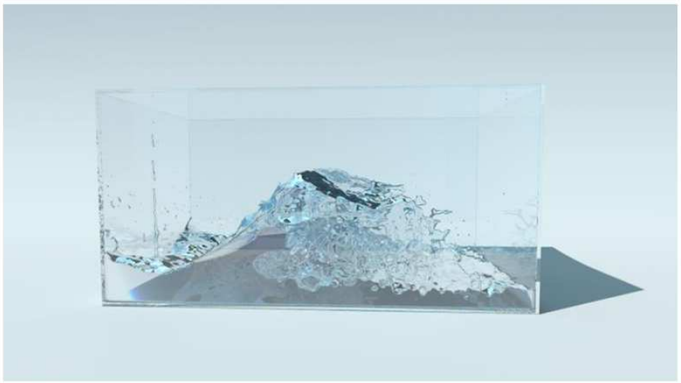


Photon Mapping (B)

- **Create an offline renderer using photon mapping techniques**
  - Capable of handling caustics from light refraction through transparent substances.
  - Accelerate photon-finding with k-d tree.
  - Solid derivation on the math behind.
- **Reference**
  - [2009] Stochastic progressive photon mapping

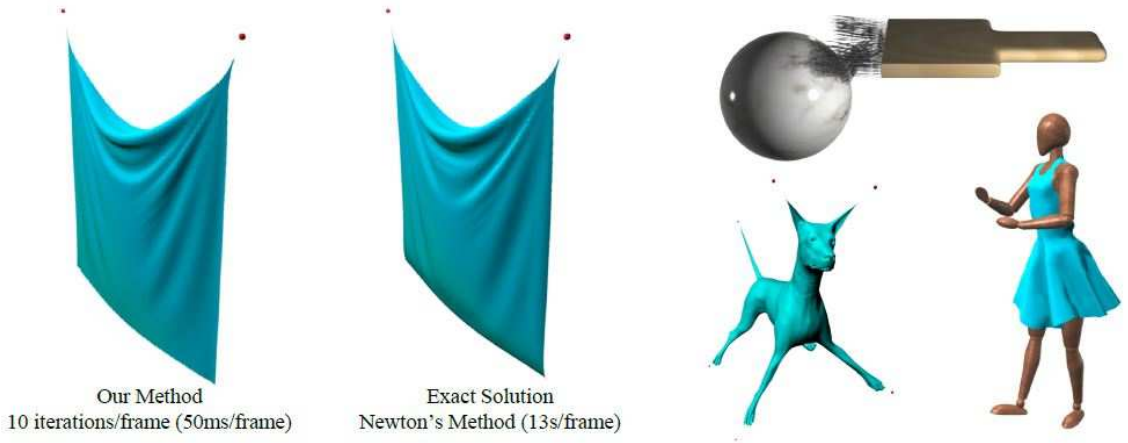
Water Surface Rendering (B)

- **A offline renderer with water surface&volume rendering capability**
  - Should handle intricate water geometry
  - Handle physically-based homogeneous volume inside the water surface
  - Can reproduce the caustics effect efficiently (select your approach)
- **Reference**
  - https://rgl.s3.eu-central-1.amazonaws.com/media/papers/Jakob2016Path.pdf
  - https://cseweb.ucsd.edu/~ravir/dilution.pdf



Cloth Simulation with Mass-Spring Model (B)

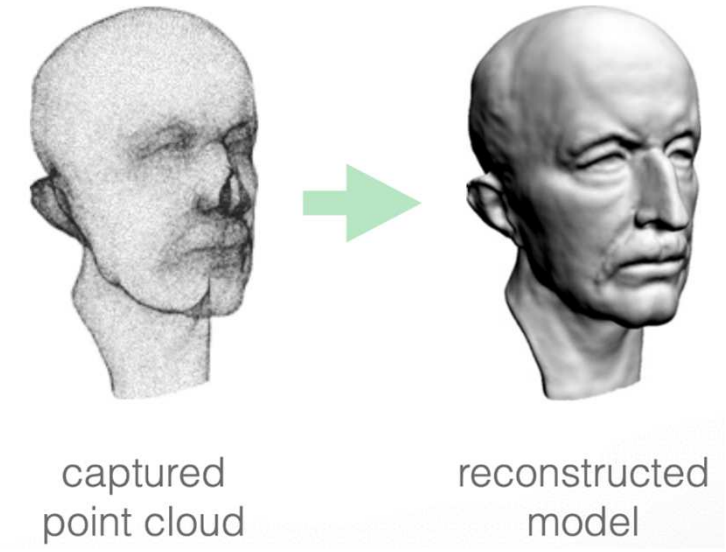
- **Fast Simulation of Mass-Spring System**
  - Implementation of the fast non-linear cloth simulation solver
  - Possibly GPU parallel implementation with CUDA
- **Reference**
  - [Fast Simulation of Mass-Spring Systems (utah.edu)]



Surface Reconstruction (B)

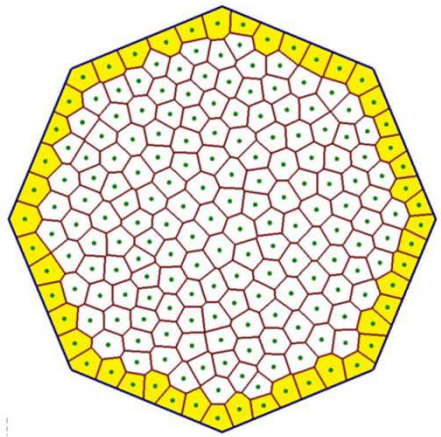
- **Poisson surface reconstruct from point clouds**
  - Estimate SDF of point clouds by Poisson equation
  - Extract meshes via marching cubes

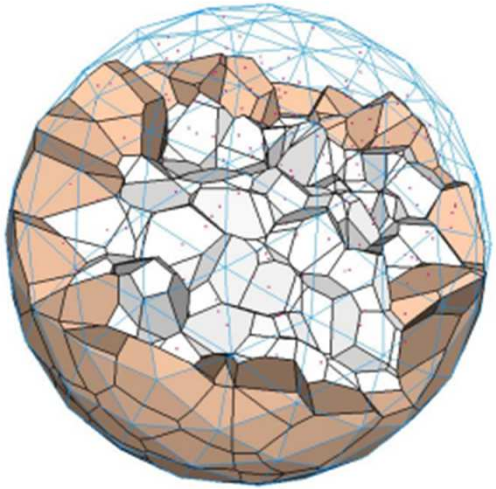
- **Reference**:
  - Michael Kazhdan, et al.
  - Poisson surface reconstruction



Voronoi Diagram (C)

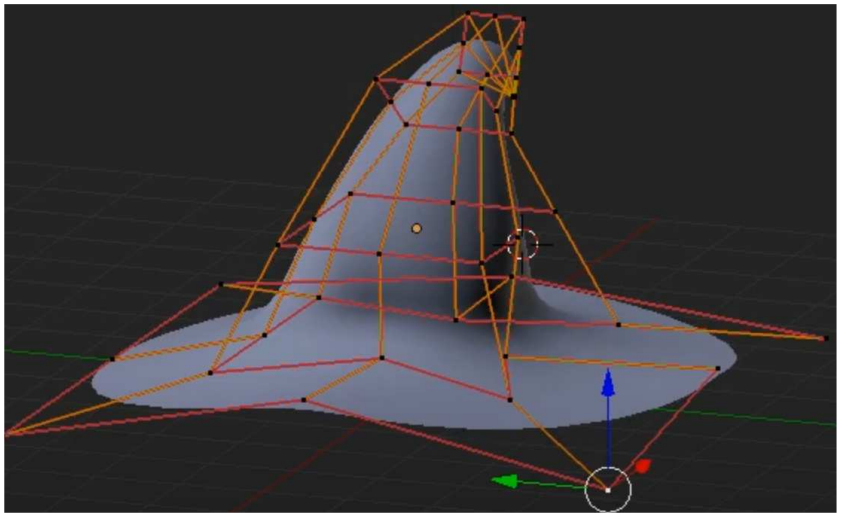
- **Compute 2D/3D clipped Voronoi Diagram**
  - Voronoi Diagram in a limited area is called clipped Voronoi Diagram.
  - Implement basic or advanced algorithm to construct Voronoi Diagram on 2D/3D erea.
  - Implement 2D: C
  - Implement a fast 3D version with fancy demo may get A
- **Reference**
  - [1986] A sweepline algorithm for Voronoi diagrams
  - [2016] Efficient Computation of 3D Clipped Voronoi Diagram
  - [2020] Parallel computation of 3D clipped Voronoi diagrams





NURBS Surface Editing (C)

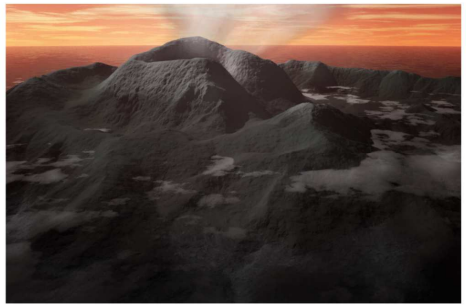
- **A small interactive NURBS modeling editor**
  - Implement NURBS evaluator
  - Create an interactive UI system for surface editing



- **Reference**:
  - Piegl L, Tiller W. The NURBS book[M].

Terrain Synthesis (C)

- **Synthesize terrain for use in computer games**
  - Synthesize terrain with stochastic heightmap or other methods
  - Free-view navigation in the generated terrain (first-person perspective).


- **Reference**
  - [2004] Realtime Procedural Terrain Generation
  - [2007] Terrain Synthesis from Digital Elevation Models
  - [2015] Parallel, Realistic and Controllable Terrain Synthesis