# Computer Graphics I

## Lecture 23: Final

**Xiaopei LIU**

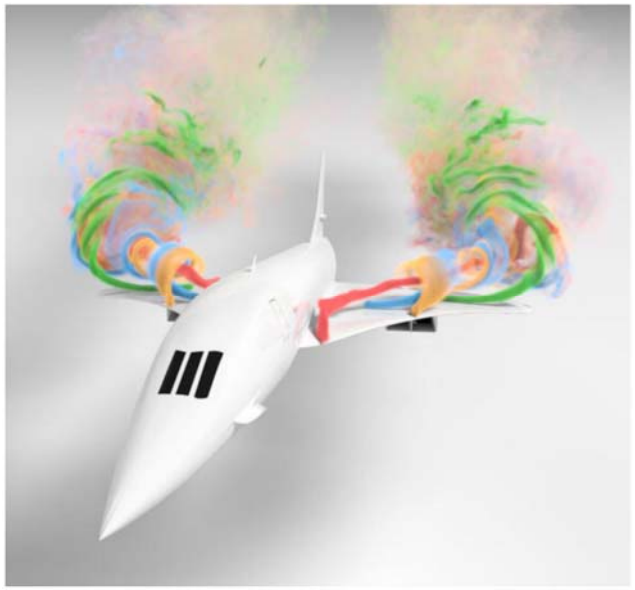
School of Information Science and Technology
ShanghaiTech University

# Real-time Smoke 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
  - Particle-based renderer using texture sprite (https://learnopengl.com/In-Practice/2D-Game/Particles)
- **Reference**: https://dl.acm.org/doi/pdf/10.1145/311535.311548 ; https://jzehnder.me/publications/advectionReflection/

# Smoke Rendering (A)

- **Render smoke by volumetric techniques**
  - Volume data are stored in VDB files
  - In-scattering effects
  - Shadow effects
  - Combined with surface rendering



- **Reference**:
  - Kutz P, et al. Spectral and Decomposition Tracking
  - Library: OpenVDB.

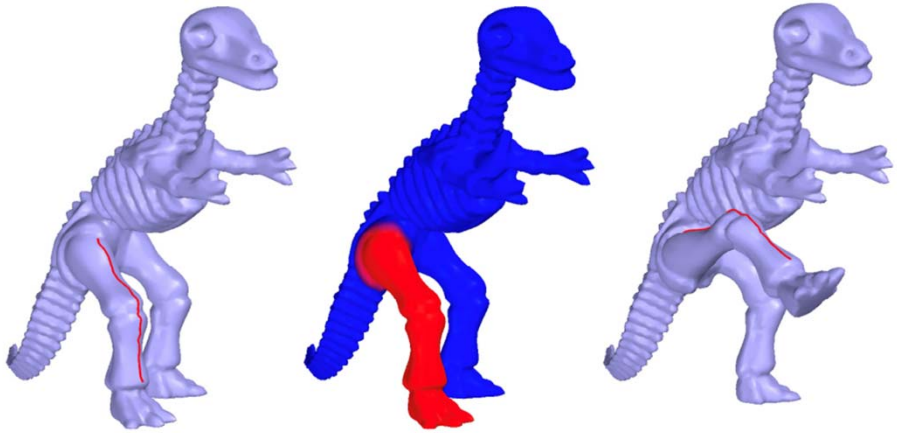
# Photon Mapping (A)

- **Create an offline renderer using photon mapping techniques**
  - Capable of handling caustics from light refraction through transparent substances.

- **Reference**:
  Henrik Wann Jensen. Realistic Image Synthesis Using Photon Mapping.
  - Matt Pharr, et al. Stochastic Progressive Photon Mapping.

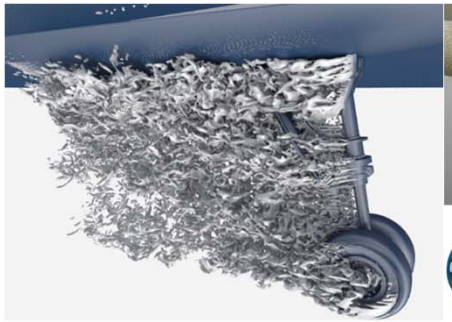
# Mesh Deformation (A)

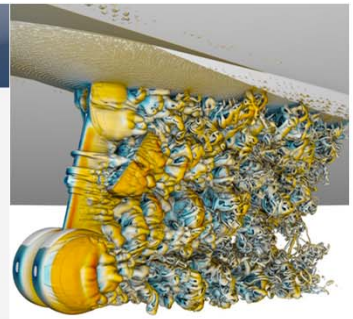
- **3D mesh deformation using volumetric graph Laplacian**
  - Design a tool to select and drag mesh curves.
  - Quadric Energy Minimization.
- **Reference**: Kun Zhou, et al. Large Mesh Deformation Using the Volumetric Graph Laplacian.



# Isosurface Rendering (A)

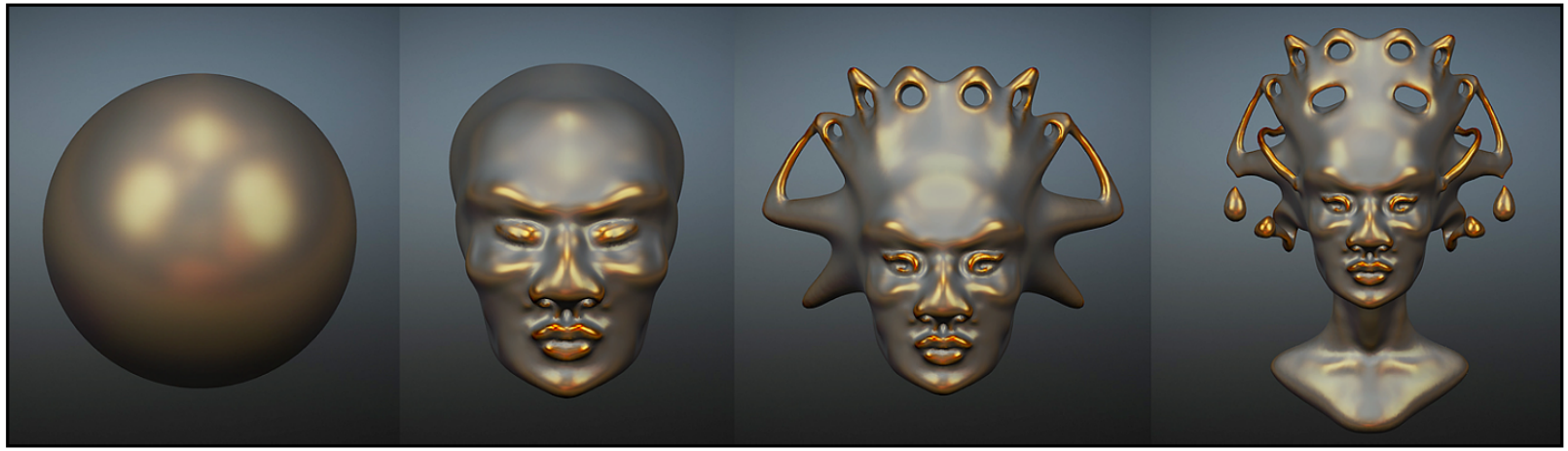
- **Render isosurfaces by volumetric techniques**
  - Build k-d trees for isosurfaces encompassed in discrete volume data
  - Design transfer functions for isosurface rendering
  - Pre-integrated transfer function
- **Reference**:
  - Interactive Isosurface Ray Tracing of Large Octree Volumes
  - High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading





# Interactive Sculpting (A)

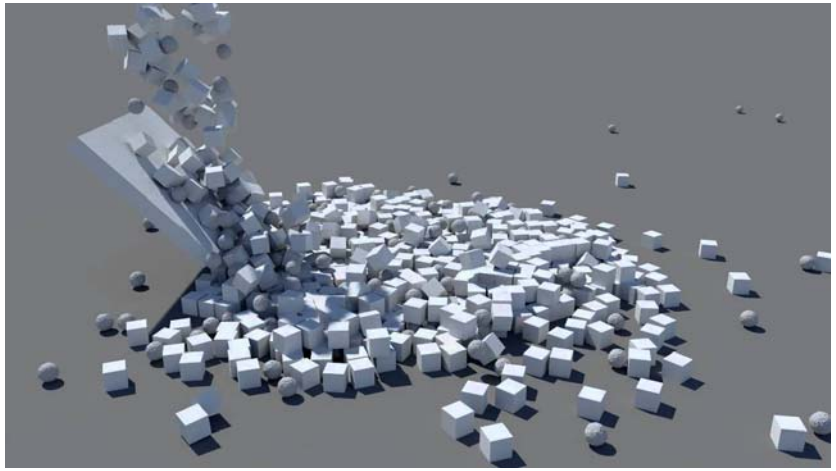
- **An interactive mesh editor that supports sculpting**
  - Interactive sculpture visual tool.
  - Dynamic changes of mesh structure.



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

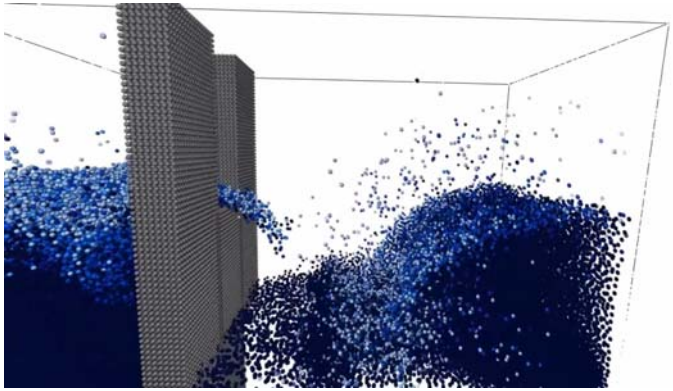
# Massive rigid-body simulation (B)

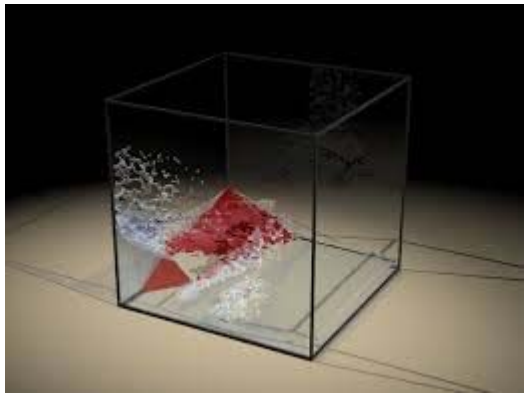
- **Implement the  massive rigid body simulation**
  - Basic transformation: translation + rotation over time
  - Collision detection
  - Handling colliding and resting contacts
- Reference:
  - Iterative Dynamics with Temporal Coherence



# Liquid simulation by SPH (B)

- **Implement the incompressible SPH solver for particle-based liquid simulation**
  - The basic SPH solver (can use existing range search library)
  - The incompressible SPH solver
  - Surface extraction for rendering or render particles
  - **Reference**: https://people.inf.ethz.ch/~sobarbar/papers/Solo9/Solo9.pdf

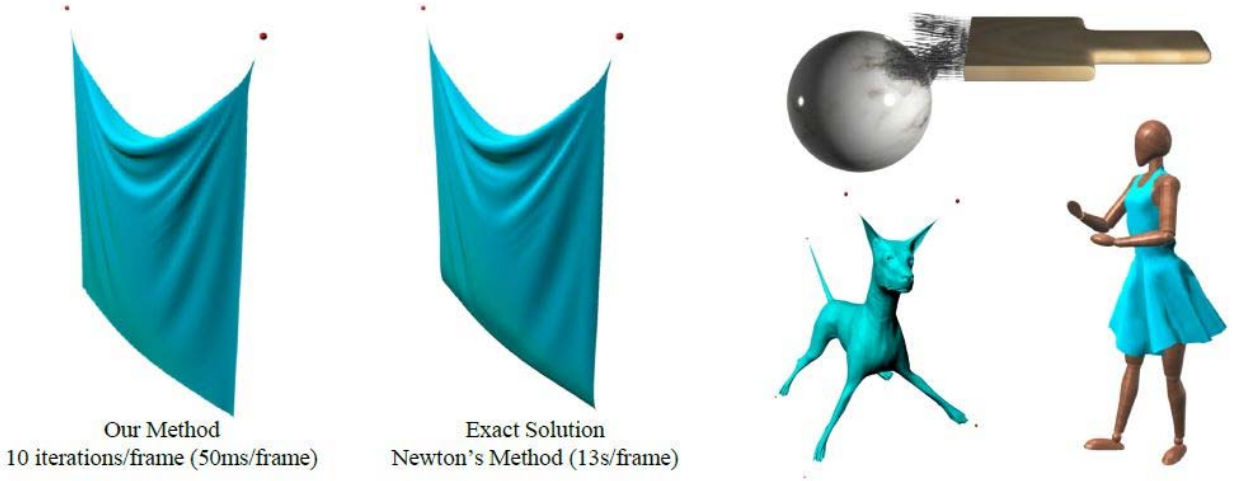




# Cloth simulation with mass-spring model (B)

- **Fast Simulation of Mass-Spring System**
  - Implementation of the fast non-linear solver
  - Possibly GPU parallel implementation

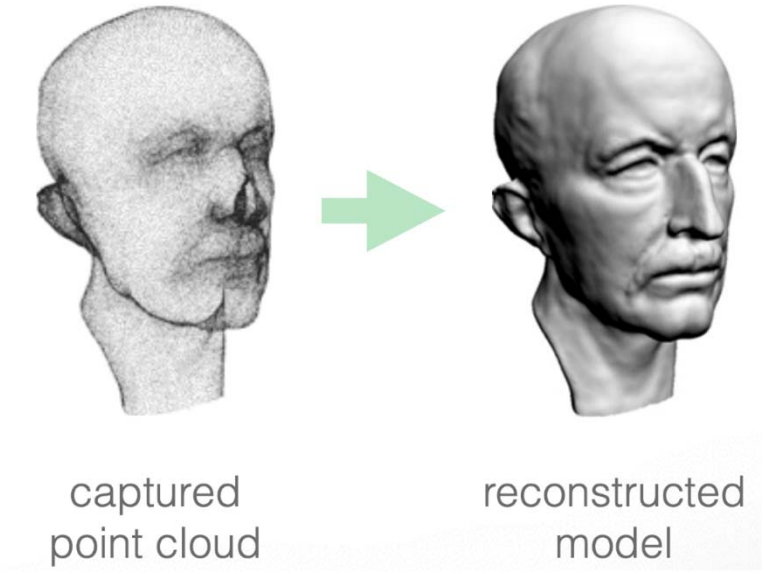
- **Reference**: https://www.cs.utah.edu/~ladislav/liu13fast/liu13fast.html



# Surface Reconstruction (C)

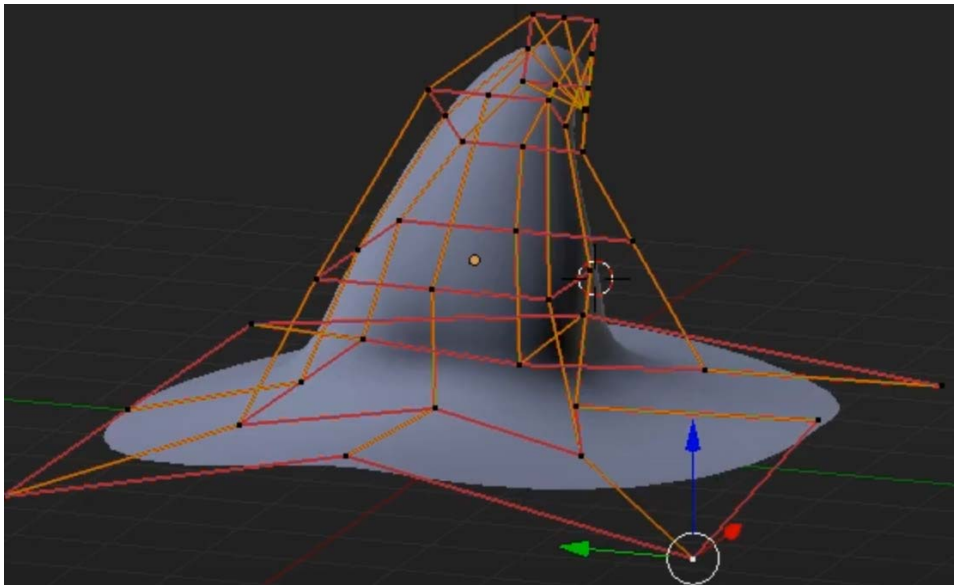
- **Poisson surface reconstruct from point clouds**
  - Estimate SDF of point clouds by Poisson equation
  - Extract meshes via marching cubes

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



# NURBS Surface Editing (C)

- **An interactive NURBS modeling editor**
  - Interactive edit NURBS.

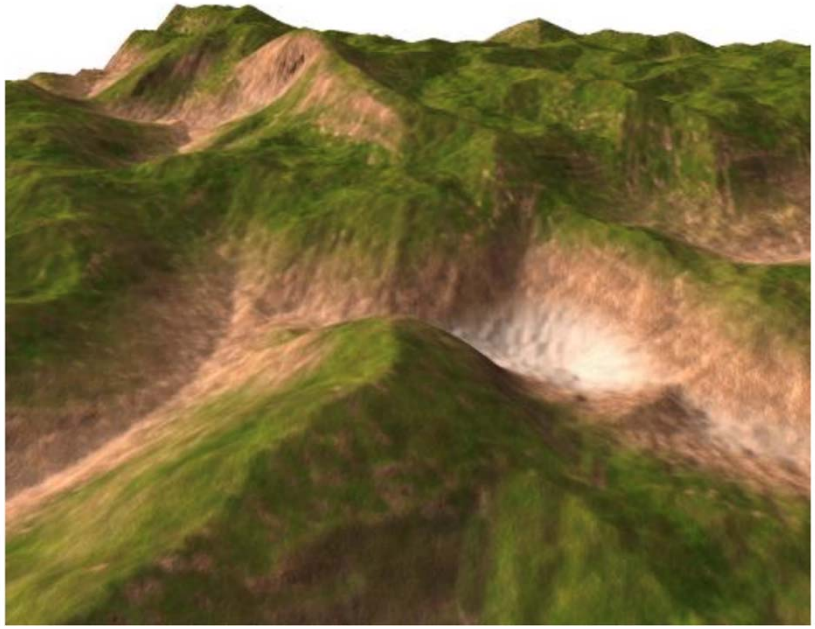


- **Reference**:
  - Piegl L, Tiller W. The NURBS book[M].

# Terrain Synthesis (C)

- **Synthesize terrain for use in computer games**
  - Generate terrain with stochastic heightmap and texture
  - Free-view navigation in the generated terrain (first-person perspective).

- **Reference**: Jacob Olsen. Realtime Procedural Terrain Generation.



# Dipole Subsurface Scattering (C)

- **Use dipole model for subsurface light transport in translucent materials**
  - Approximate BSSRDF through dipole model.
  - Optional: implement directional dipole model.
- **Reference**: Henrik Wann Jensen, et al. A Practical Model for Subsurface Light Transport.