Realistic and Interactive Simulation of Rivers

Peter Kipfer

Havok

peter.kipfer@havok.com

Rüdiger Westermann

Computer Graphics & Visualization
Technische Universität München
westermann@in.tum.de

Motivation

- Interactive physically-based simulation of water flowing on a terrain
  - Quick flow analysis
  - Simulator environments
  - Games
- Challenges
  - Simulation
    - Full particle interaction
  - Surface extraction
    - Fast processing
    - Static memory footprint
    - Explicit representation

Smoothed particle hydrodynamics

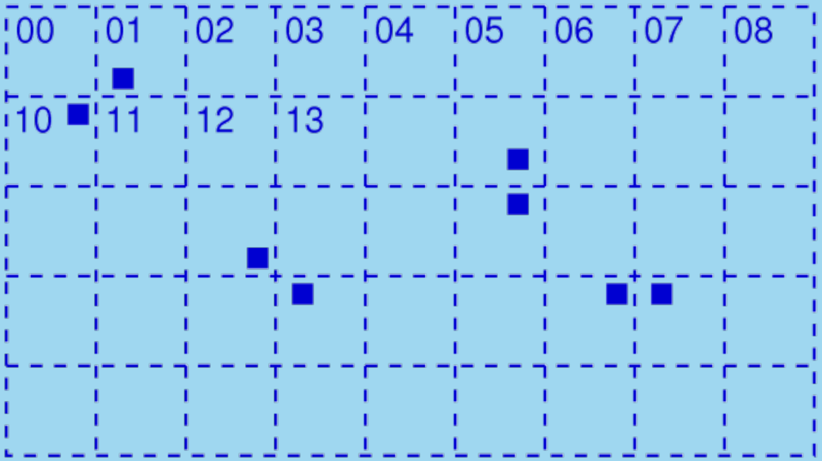
- Lagrangian approach for fluid dynamics [Müller2003]

$$A_S(\vec{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\vec{r} - \vec{r}_j, h)$$

- Interpolate quantity A using smoothing kernel W
- Particle neighborhood queries
- Hierarchical search methods
- Octree-based approach [Vemuri1998]
  - Linear complexity if bin size = 2*(max. particle diameter)
  - Dynamic memory allocation necessary
  - Sensitive to particle count and density

Neighborhood by linear search

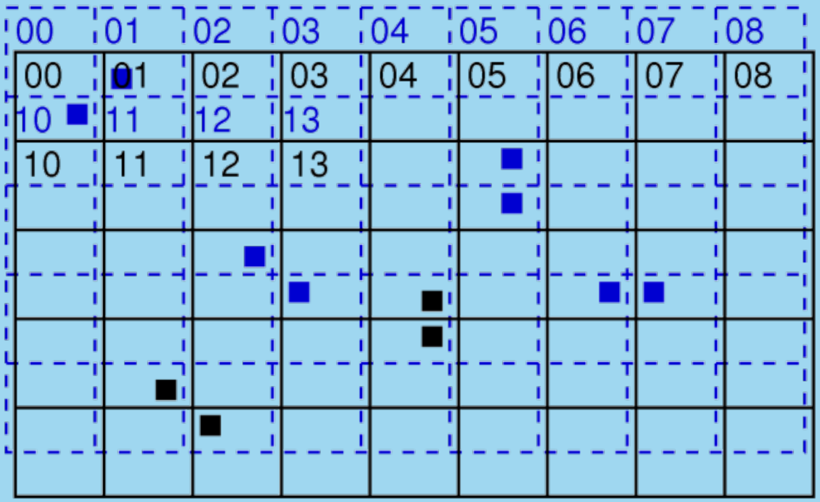
1. Quantize domain to bins
2. Build particle id from bin numbers: $0|i_z|i_y|i_x$
3. Sort according to id
4. Collide with all left neighbors until $id \leq (id_{self} - 2)$



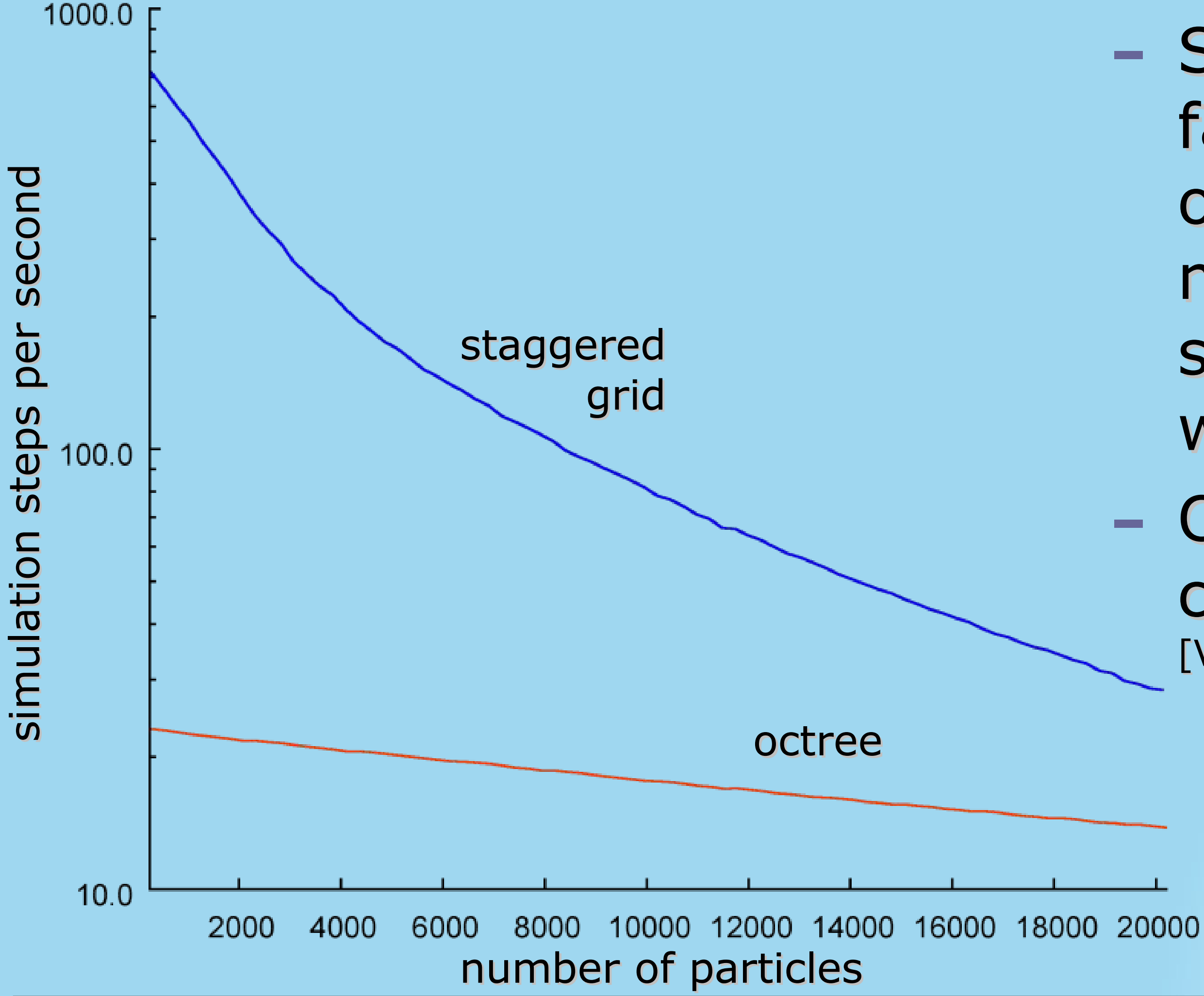
- Cache efficient
- Linear time complexity for traversal
- Sort complexity O(n log n)

Staggered grid

- For arbitrary dimensions
  - Add one staggered grid for every missing dimension
  - Use leading bit per dimension processed to avoid redundant collisions
- Finds all neighbors
- Linear traversal (fixed bin and fixed max. particle size)
- Static memory footprint
- Insensitive to particle count and density



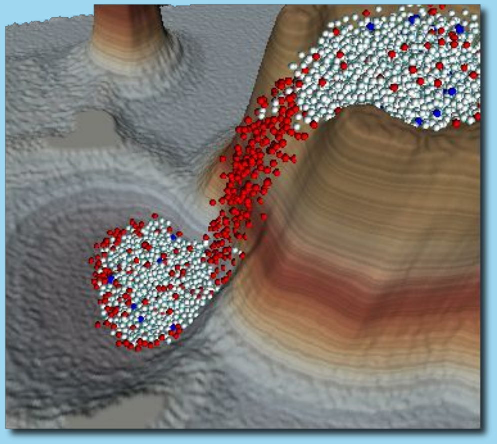
Performance

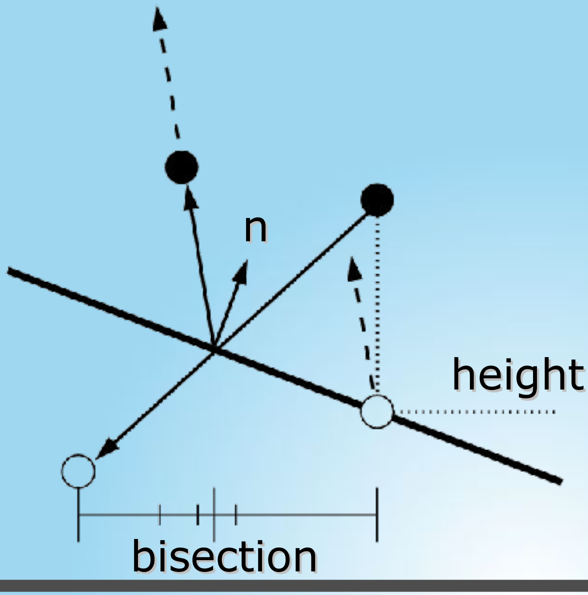


- Staggered grid is faster by an order of magnitude for sparse systems we target
- Octree has huge constant factor [Vemuri1998]

Optimized collision resolution

- Introduce „frozen“ particles
  - Exclude from terrain collision
- Precision terrain collision avoids motion artifacts
  - Iterative backtracking of motion path (bisection)
  - Inelastic bounce using surface parameters
  - Recompute path length accounting for velocity damping of collision

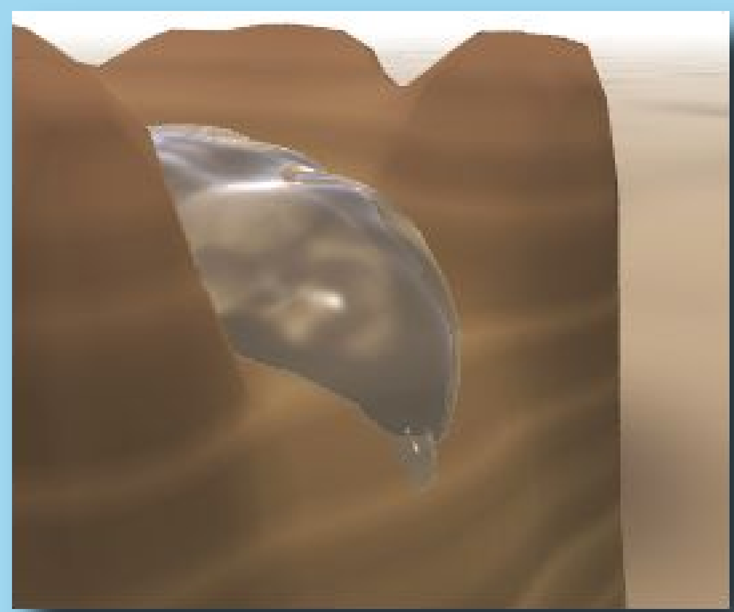




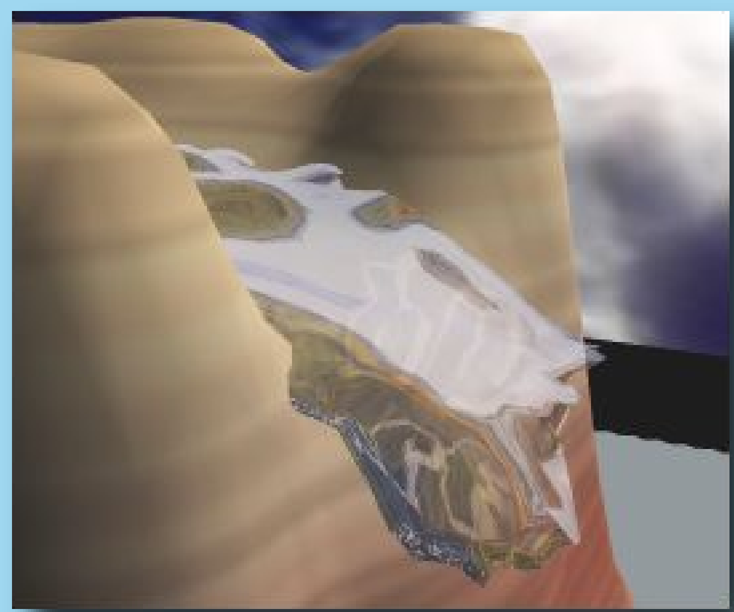
Surface extraction

- Marching cubes or raycasting
  - Construct fluid free surface as level set of particle radii
  - Very high tessellation / small integration step size needed for detail
  - High memory requirements / sampling artifacts
  - Incremental update very difficult: no GPU storage
- Point splatting
  - Fast
  - Dense particle set needed
  - No explicit surface representation
  - High quality splats (surfels) need additional data per particle

Appearance



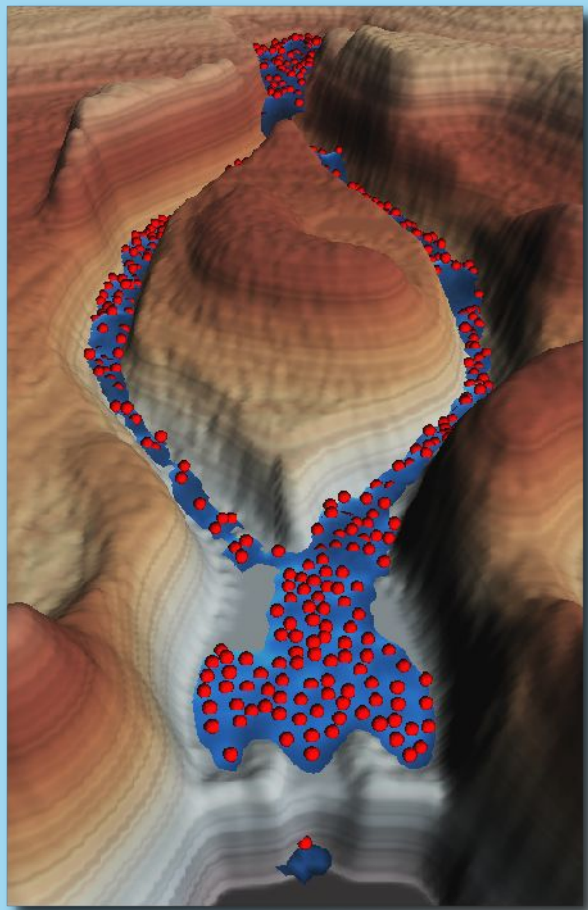
raytraced surface



carpet surface

The carpet

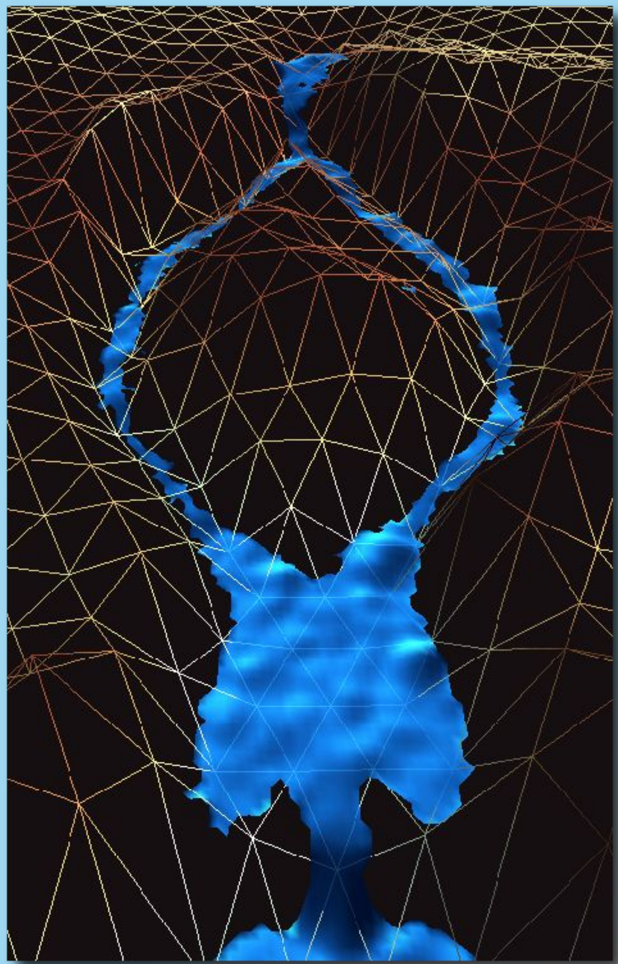
- Interactive display
  - Visualize every simulation step
- Sparse particle set
  - Closed surface
- Large terrains
  - Generate only where particles are
- Explicit surface representation
  - Advanced rendering shaders
- Split / join handled automatically



CPU carpet construction

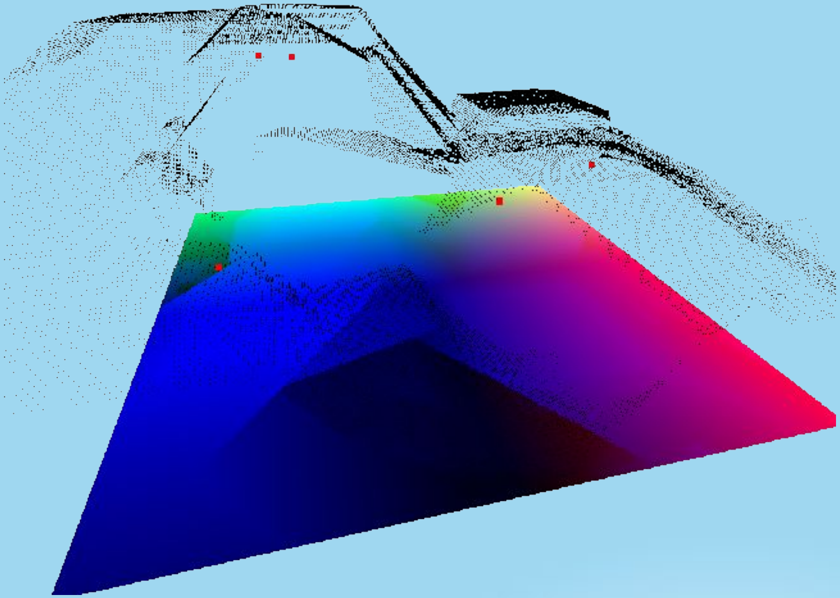
1. Store all particles in quadtree leaves
2. Pull particle height up the tree using max operation
3. Top-down traversal to leaf
   a) render quad
   b) accelerate position downwards

- Constant memory footprint
- Linear complexity
- Multi-threaded
- Step 1 folded into advection



GPU carpet construction

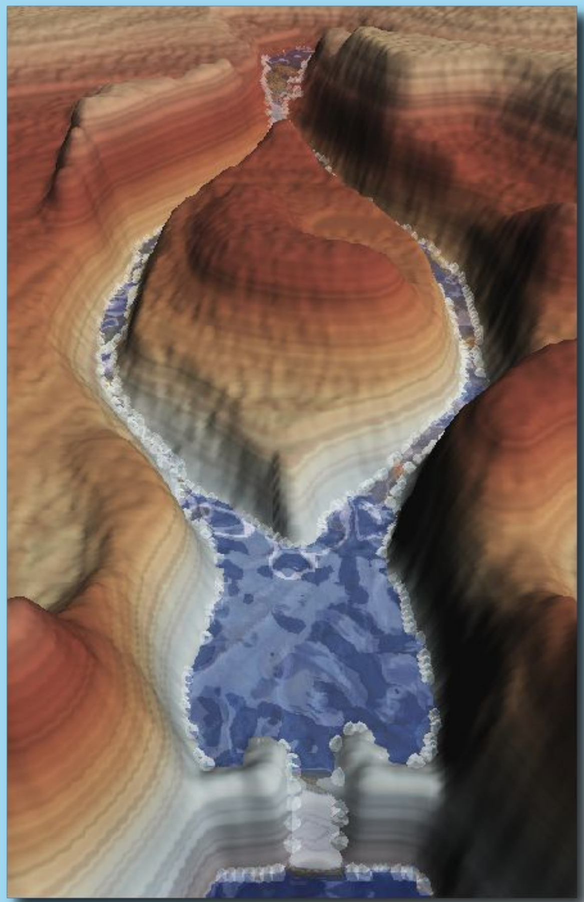
1. Create z-image and apply gravity to carpet vertices
2. Splat particles to depth image with z-test on
3. Render vertex array as triangle strip



- Minimal data upload
- Incremental update
- Provide shape for particle using oriented depth sprites
- Efficient rendering possible (early z-test)

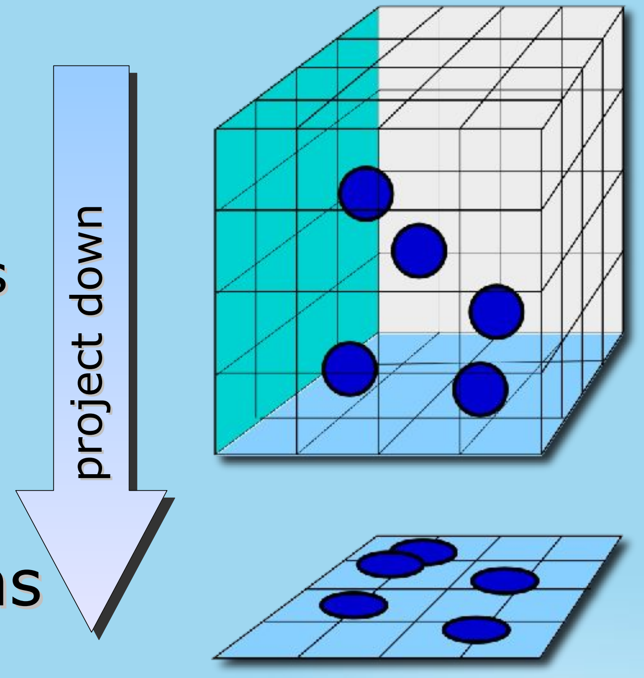
Carpet rendering

- Augmented rendering by using provided per vertex information
  - Flow direction
  - Local speed
  - Water depth
- Surface waves / wind
  - Procedural gravity update
- Influence simulation
  - Terrain surface adhesion
  - Friction modulation („Wet map")
  - Erosion

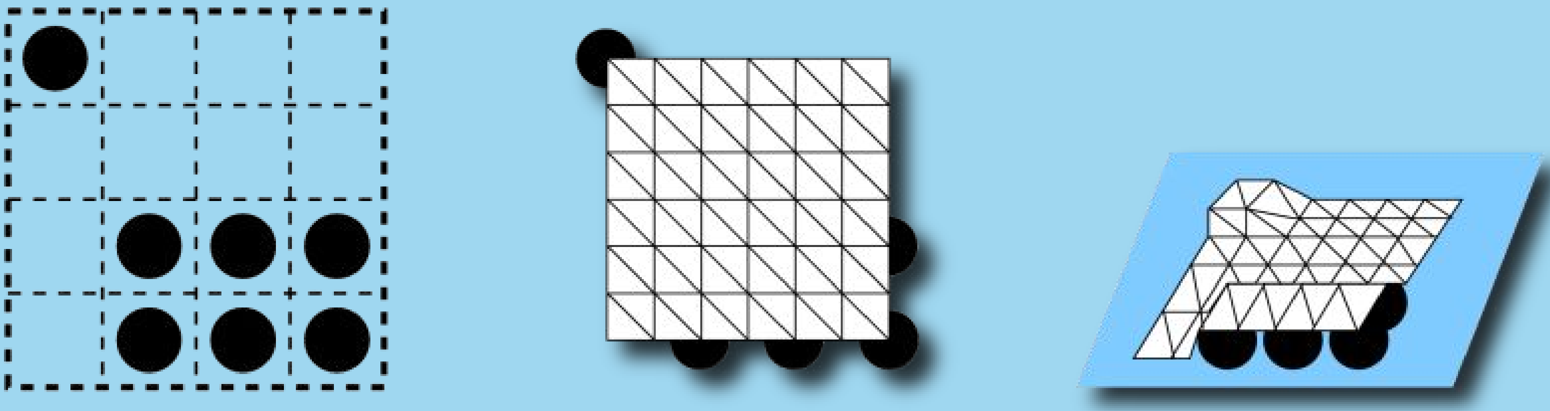


Carpet optimization

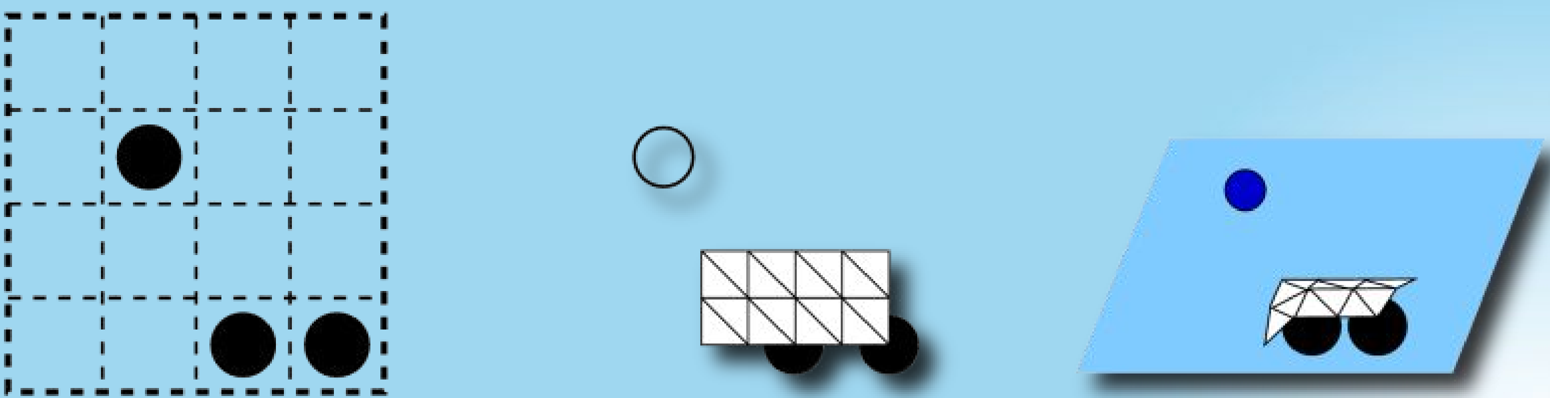
- High resolution carpet or very large terrain
  - Truncate quadtree
  - Leaves contain a set of carpet bins
  - Projection is very simple (mask on particle id)
- Store resulting surface of pre-computed particle configurations
  - High quality surface tension
  - Different rendering primitives for isolated parts
  - LOD



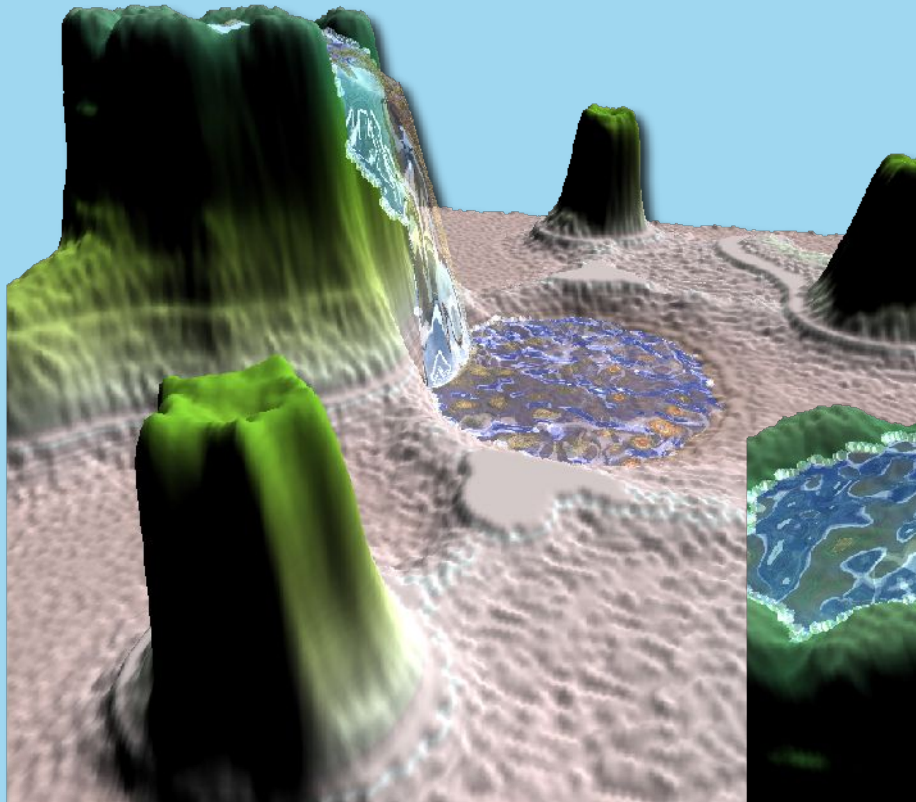
Configuration examples

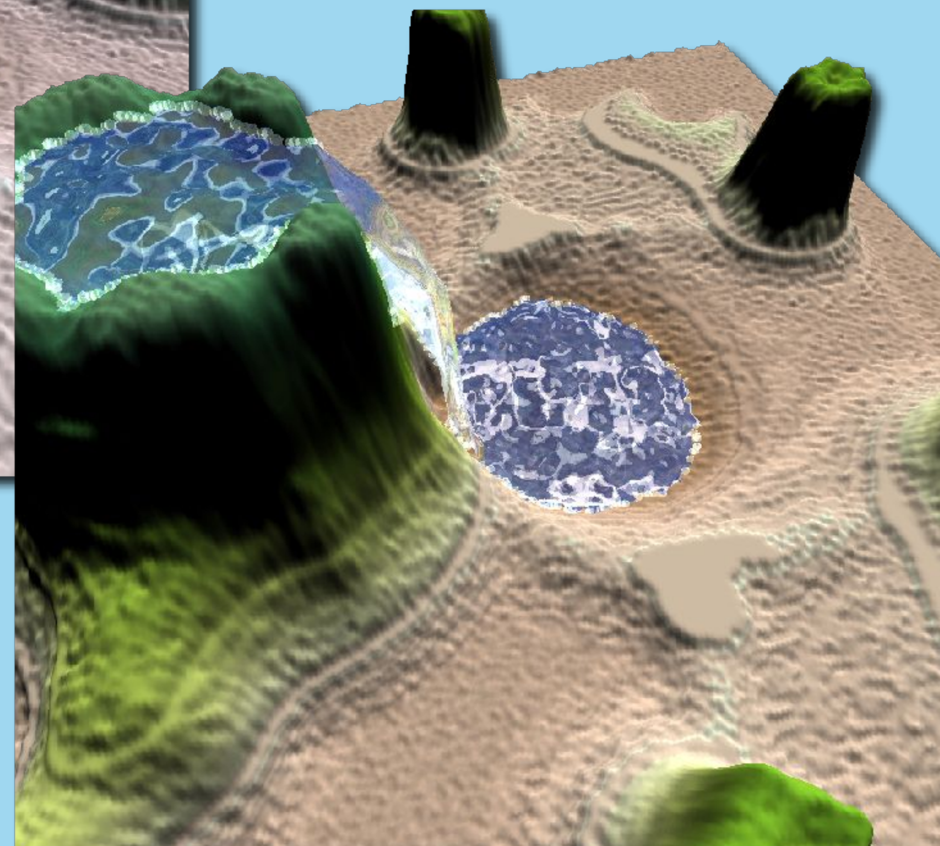


projected configuration | dual carpet quadtree mesh | rendered carpet (side view)



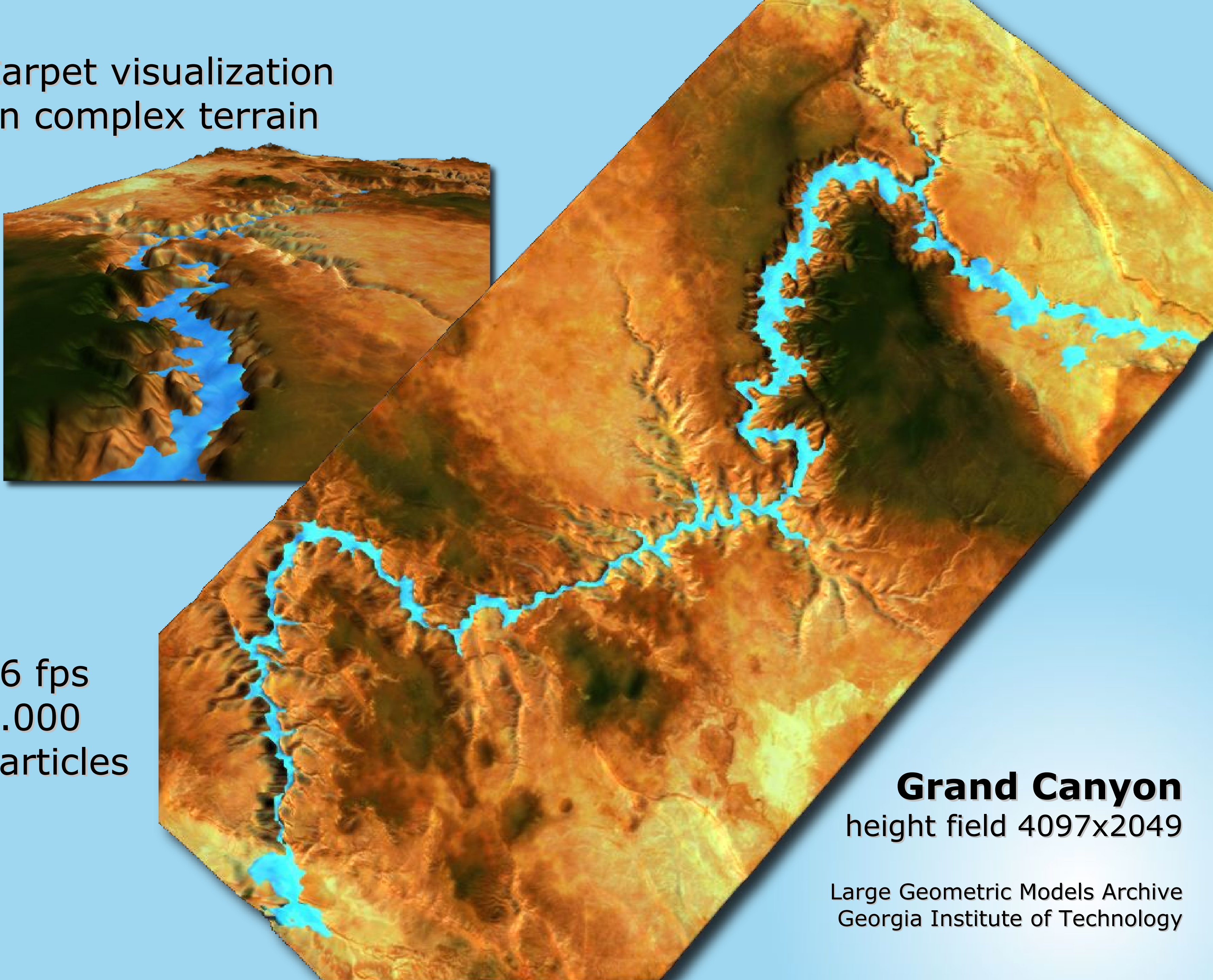
Carpet visualization for real-time applications in extreme cases





68 fps
3.000 particles

Carpet visualization on complex terrain

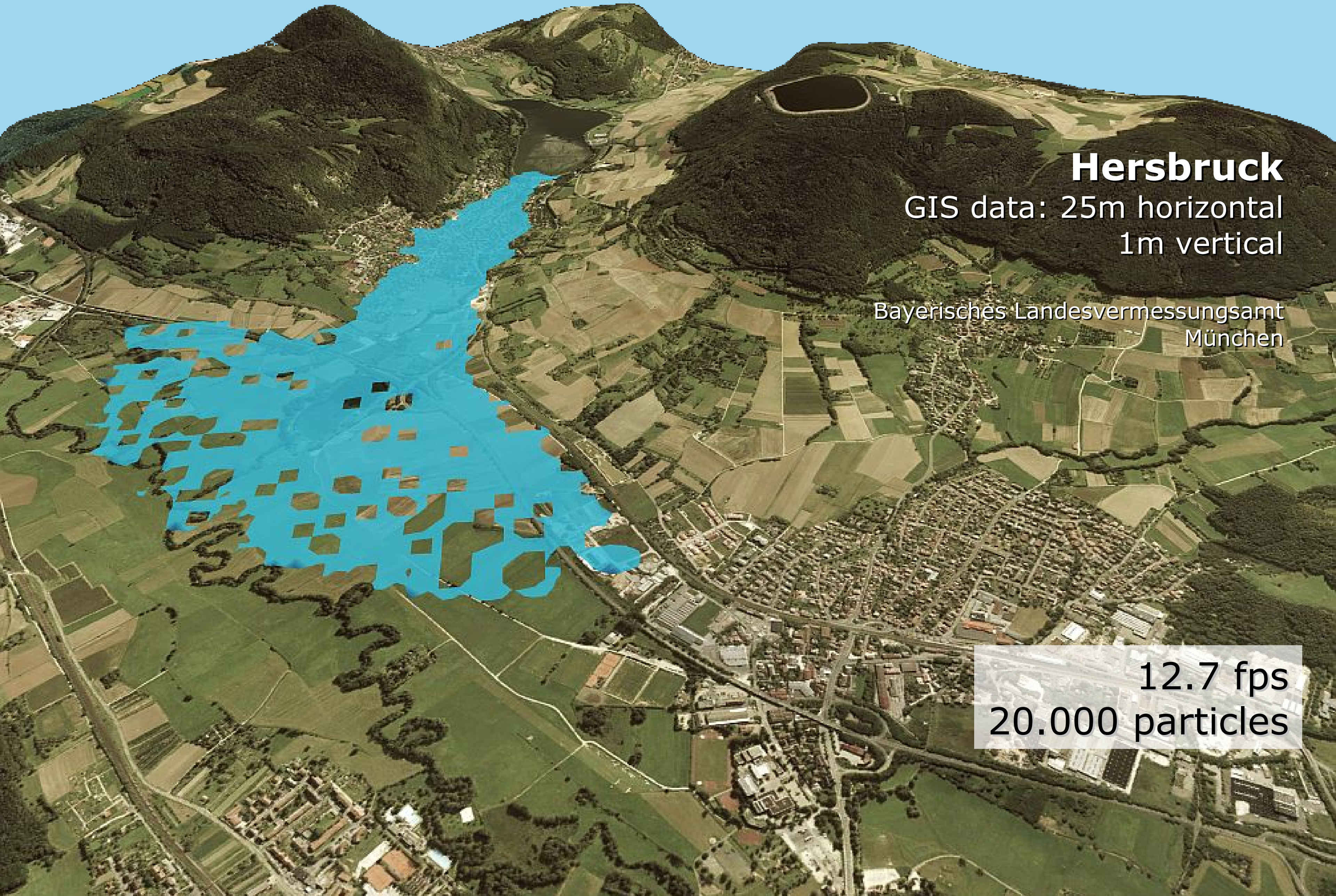


26 fps
8.000 particles

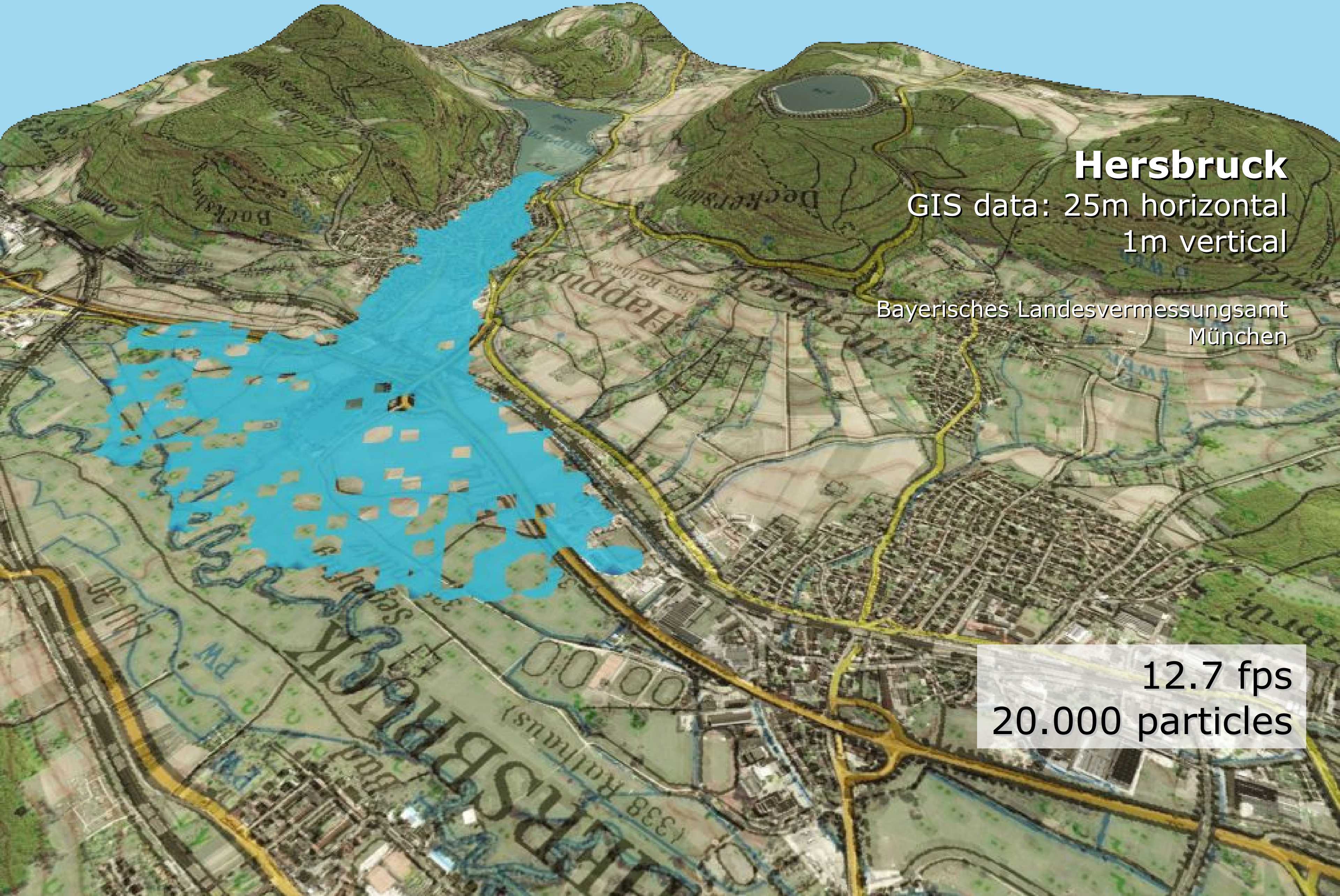
Grand Canyon
height field 4097x2049

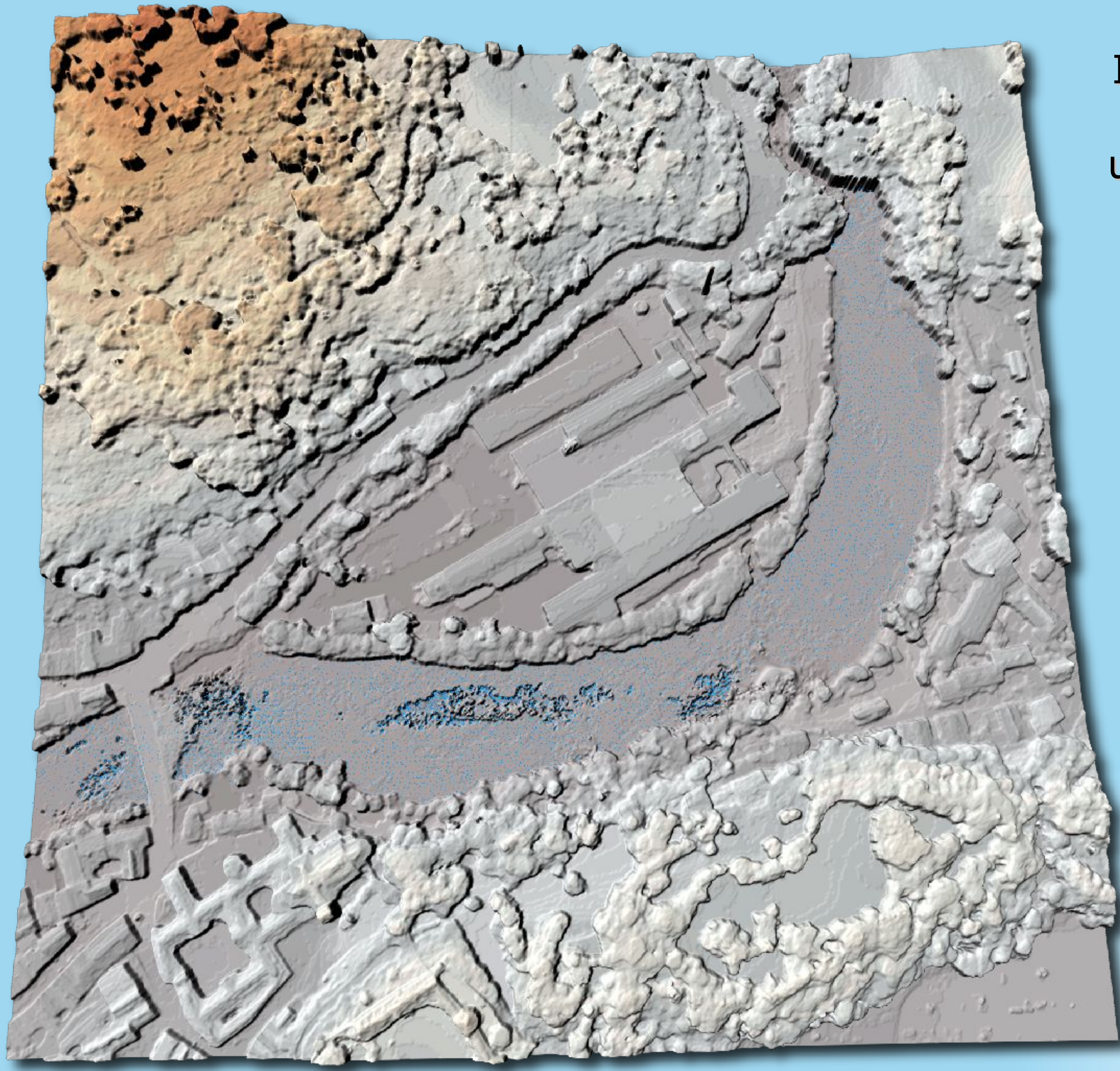
Large Geometric Models Archive
Georgia Institute of Technology

Carpet visualization for environmental simulation



Carpet visualization for environmental simulation



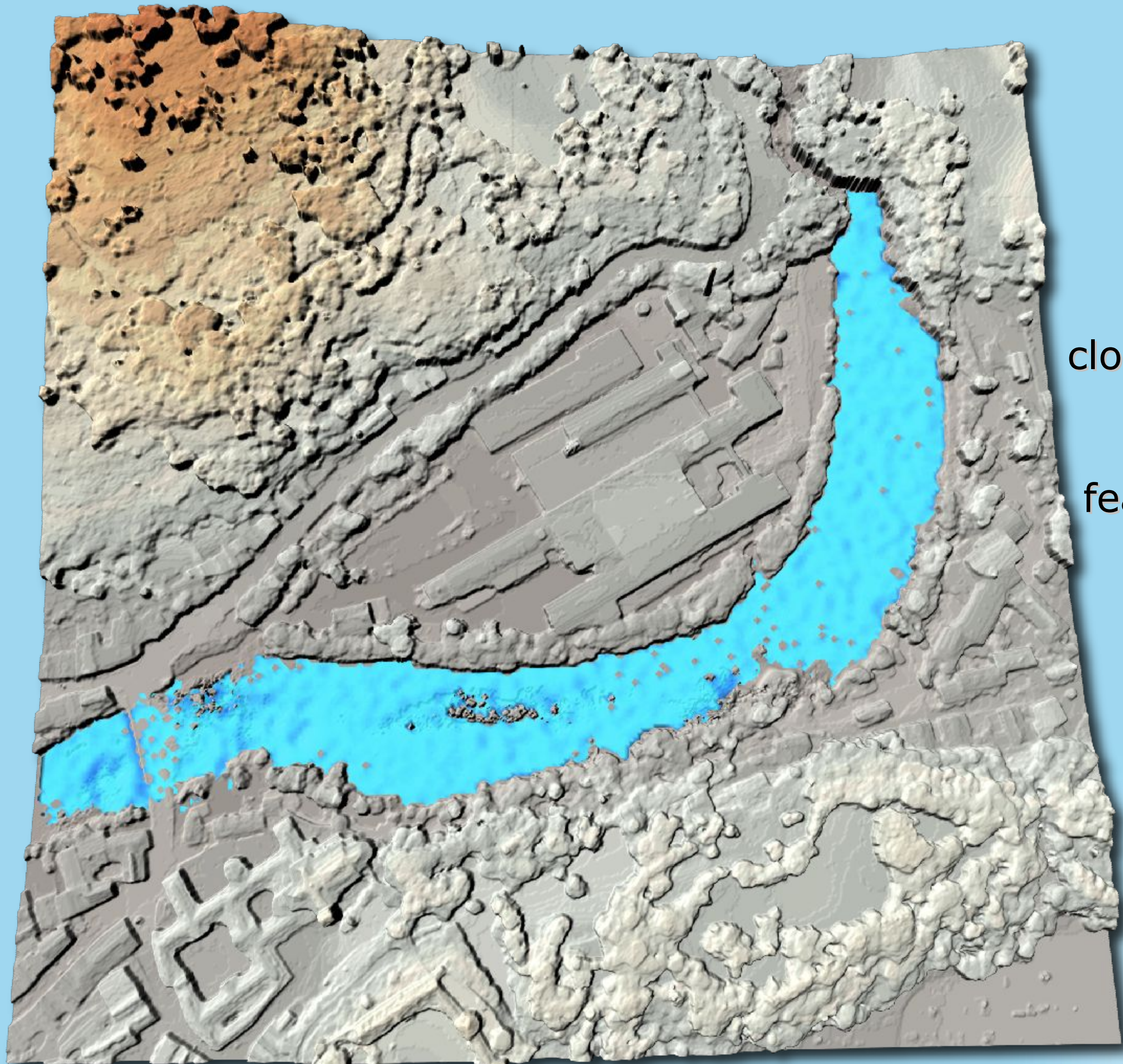


Instant visualization and simulation on ultra-high resolution scan data

HRSC camera data from single-pass overflight

Resolution:
15 cm horizontal
10 cm vertical

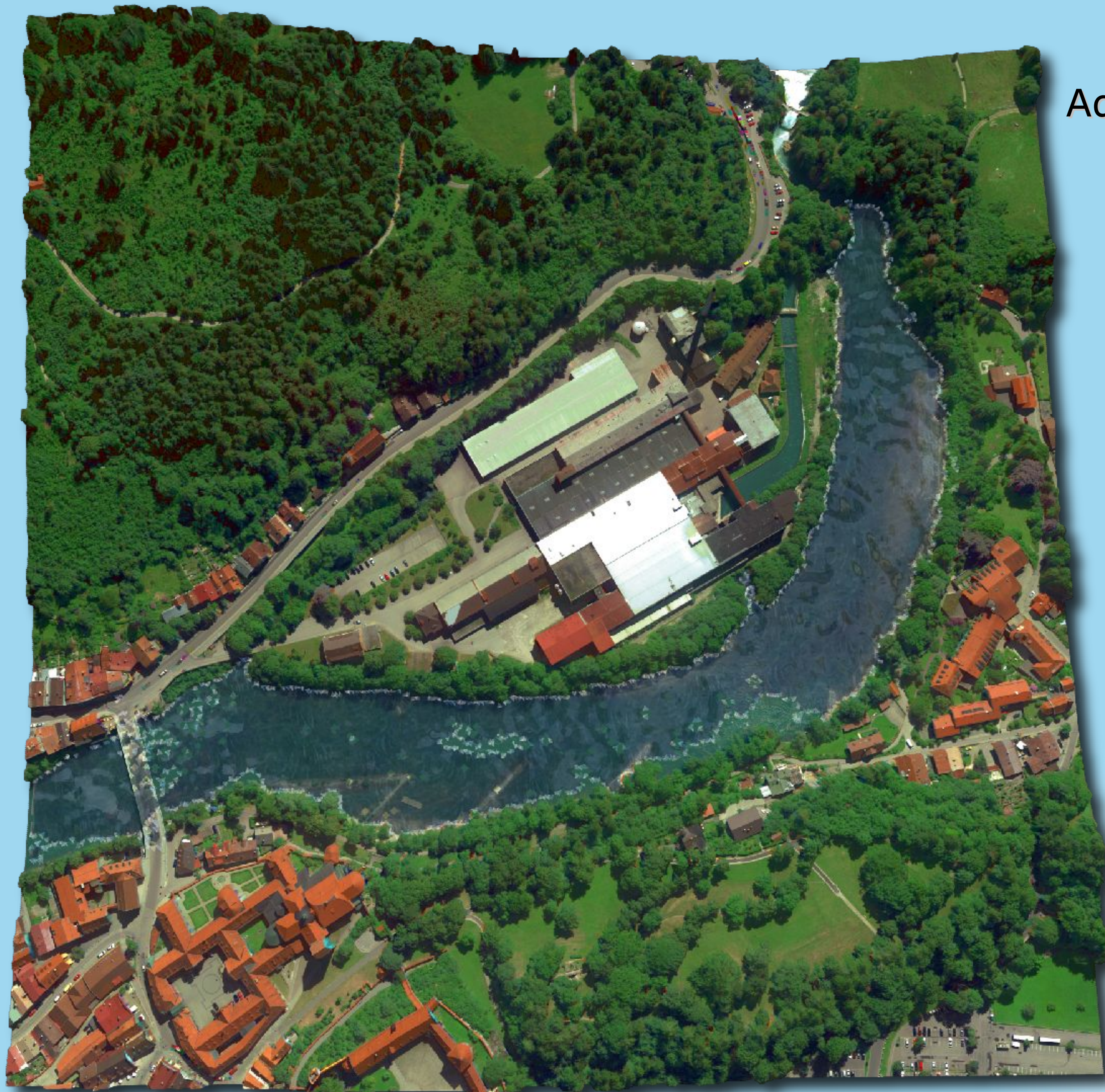
German Aerospace Center (DLR)
Institute of Robotics and Mechatronics



Highly interactive simulation of river behavior

Carpet provides closed surface despite sparse particle set without hiding flow features or obscuring terrain detail

German Aerospace Center (DLR)
Institute of Robotics and Mechatronics

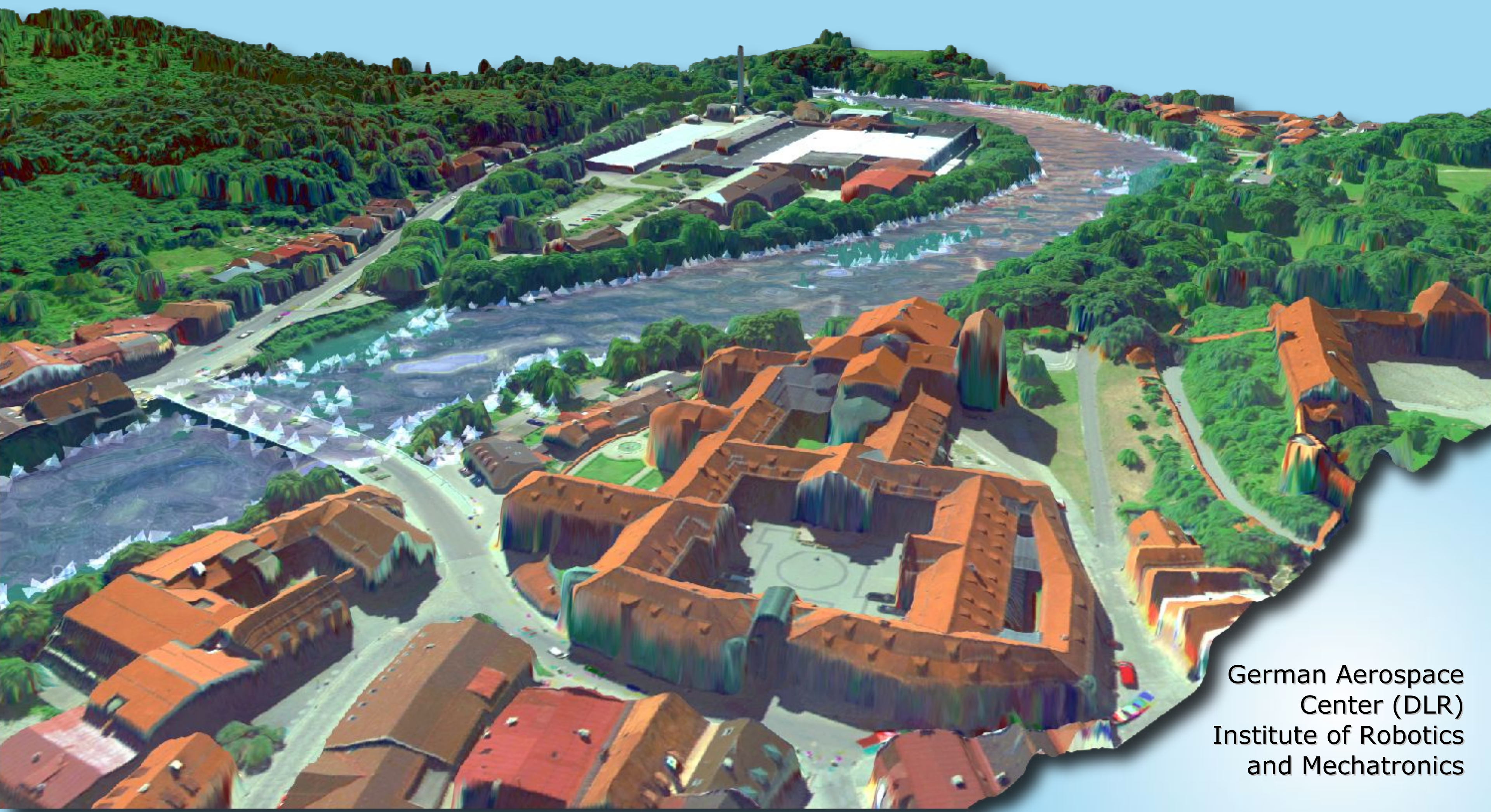


Add textures using the chromatic HRSC channels

German Aerospace Center (DLR)
Institute of Robotics and Mechatronics

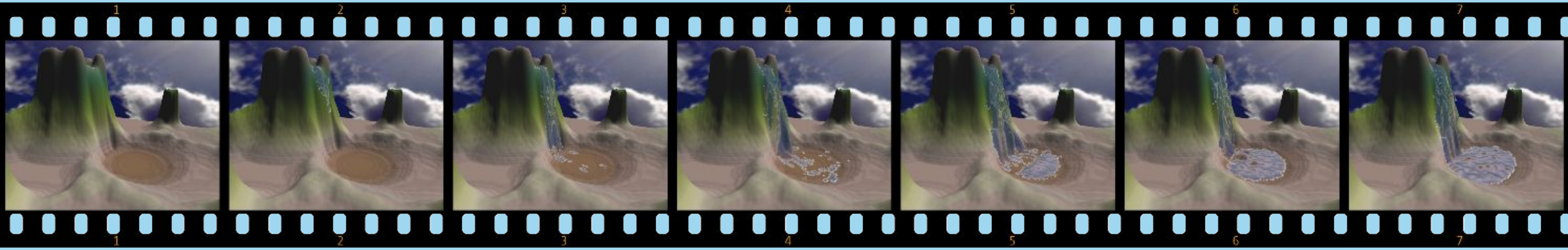
21 fps
9.500 particles

Füssen
HRSC scan
15 cm horizontal
10 cm vertical



German Aerospace
Center (DLR)
Institute of Robotics
and Mechatronics

Video



Demo + Infos

http://wwwcg.in.tum.de/Research/Publications/RiverSim