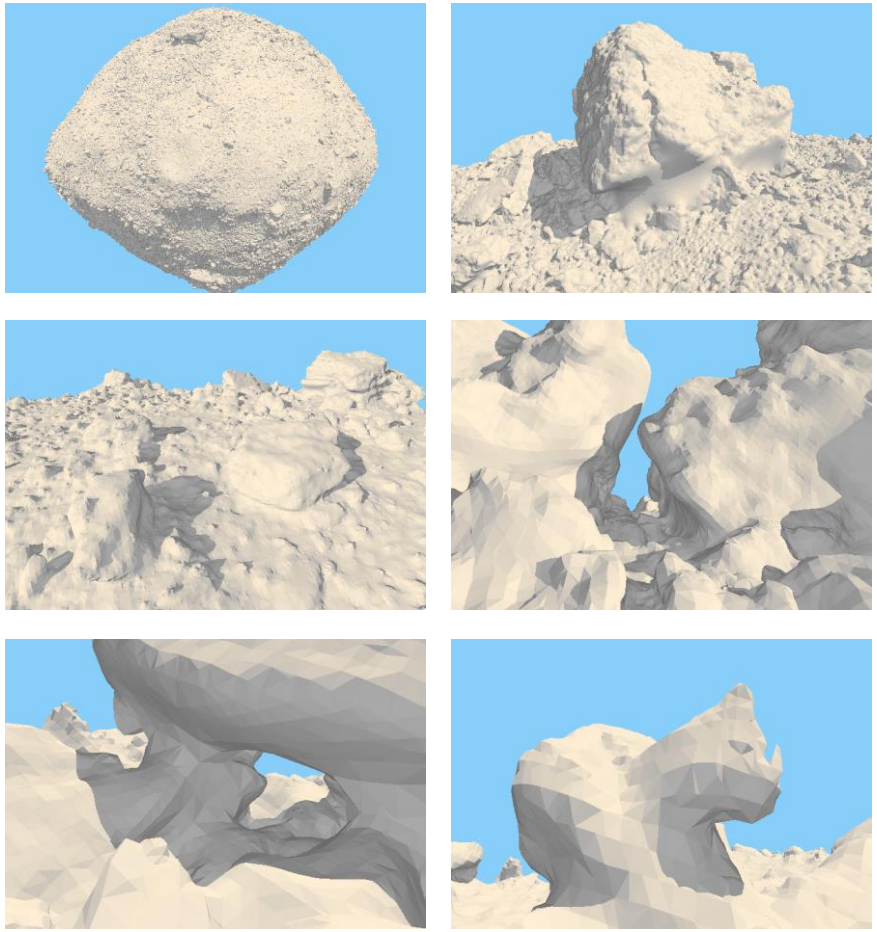
SIGGRAPH 2021

MODELING ASTEROID (101955) BENNU AS A REAL-TIME TERRAIN

MATTHIAS ENGLERT / TINMAN 3D / GMT+1

ASTEROID BENNU (101955)



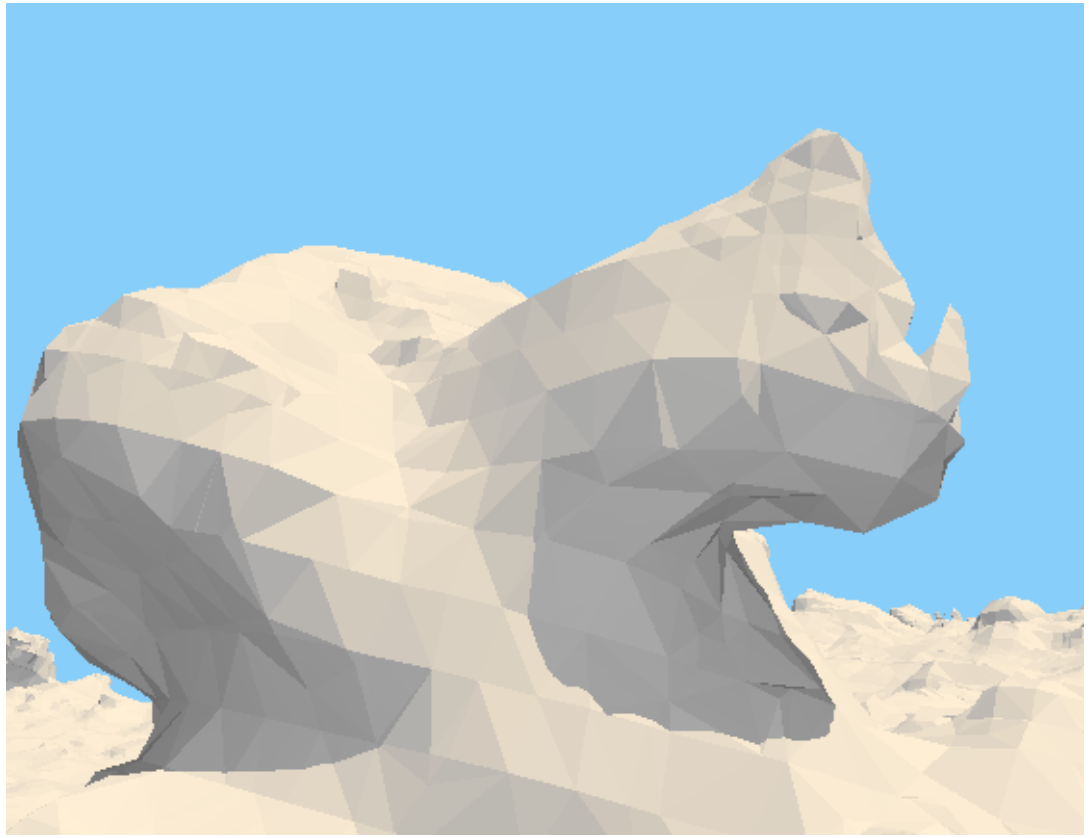
- **NASA OSIRIS-REx mission /** https://www.nasa.gov/osiris-rex
  “OSIRIS-REx traveled to near-Earth asteroid Bennu and is bringing a small sample back to Earth for study. The mission launched Sept. 8, 2016, from Cape Canaveral Air Force Station. The spacecraft reached Bennu in 2018 and will return a sample to Earth in 2023.”
- This included a **detailed survey** of Bennu, followed by the generation of a **highly-detailed 3D model:**
  - OBJ file format
  - Approx. 8 GB file size
  - Approx. 177,000,000 triangles
- **Thanks, NASA!**

LOADING BENNU

TRYING TO LOAD THE OBJ FILE...

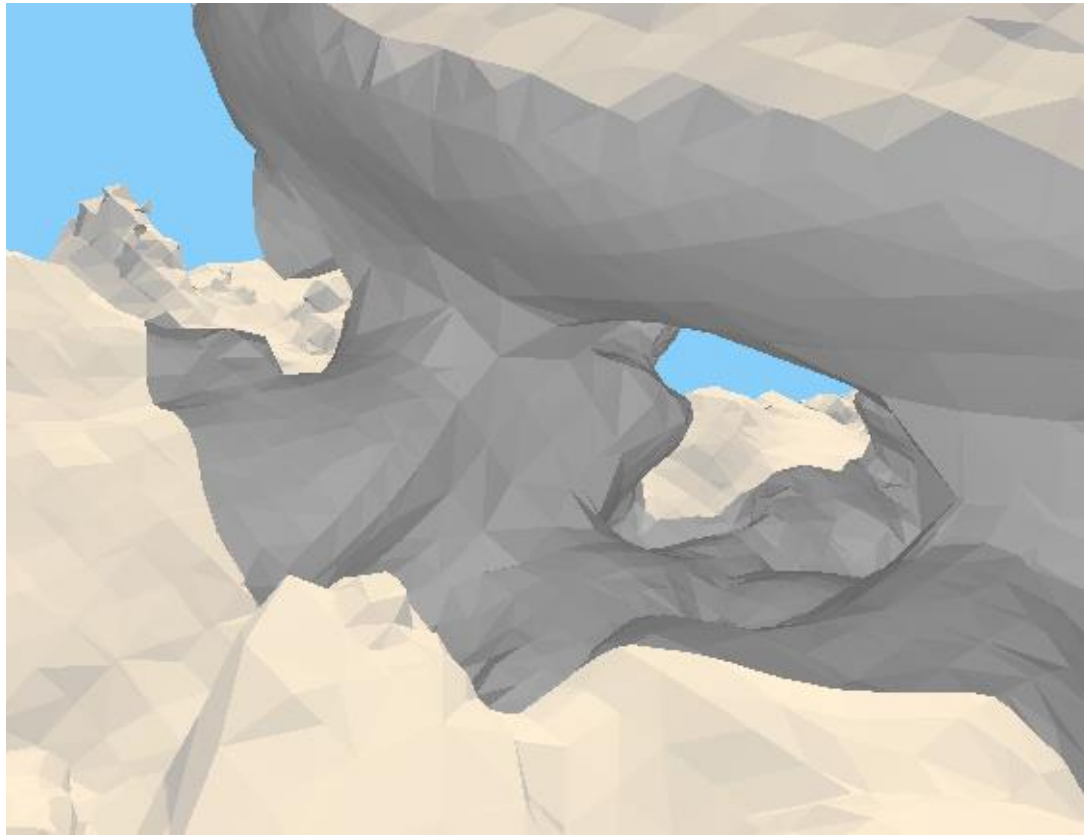
- …is challenging, as standard tools are not always designed to handle huge amounts of geometry.
- …is tedious, because many 3D model processing steps are not O(N), making you wait (forever).
- …is suspenseful, because you never know whether the tool is still working or has crashed or got stuck.
- …finally succeeded with the Tinman 3D SDK, after fixing a whole lot of source code issues – *Phew!*

RENDERING BENNU



GROUND TRUTH

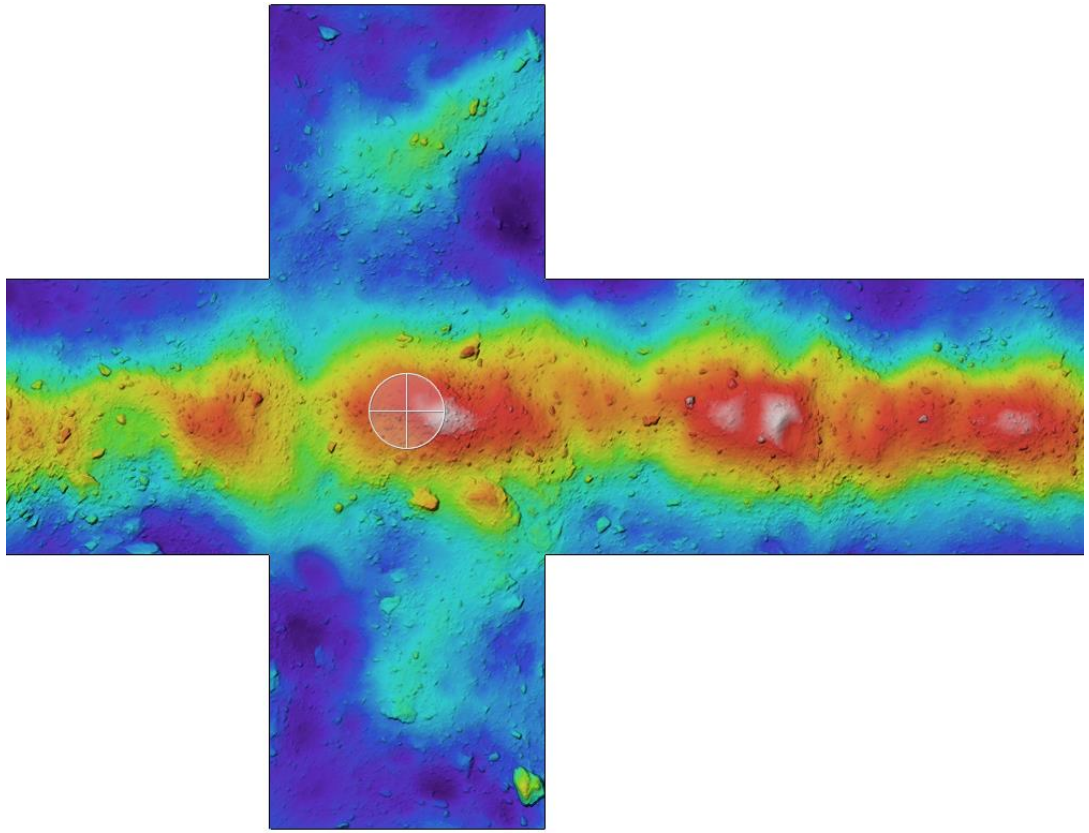
- Need “Ground truth” to validate results later
- **Naïve brute-force means…**
  - …to shovel ~177E+6 unique tris onto the GPU
  - …and to render all of them each frame.
- **Better this (for example):**
  - Build spatial acceleration structure (octree)
  - Perform ray-casting / ray-tracing (see left)
- Scientific datasets are accurate but also “ugly”.
- **How to make this more appealing?**



MAKING A BENNU TERRAIN

- **Why create a 2.5D terrain for Bennu?**
  - Make use of existing terrain technologies: Heightmap + Displacement + LOD system
  - Apply common texturing techniques
  - Sacrifice scientific accuracy (for aesthetics)
  - Try best to capture overhangs / holes / caves
- **We need:**
  - An elevation map
  - A displacement map
  - A material splat-map (see demo code for details)

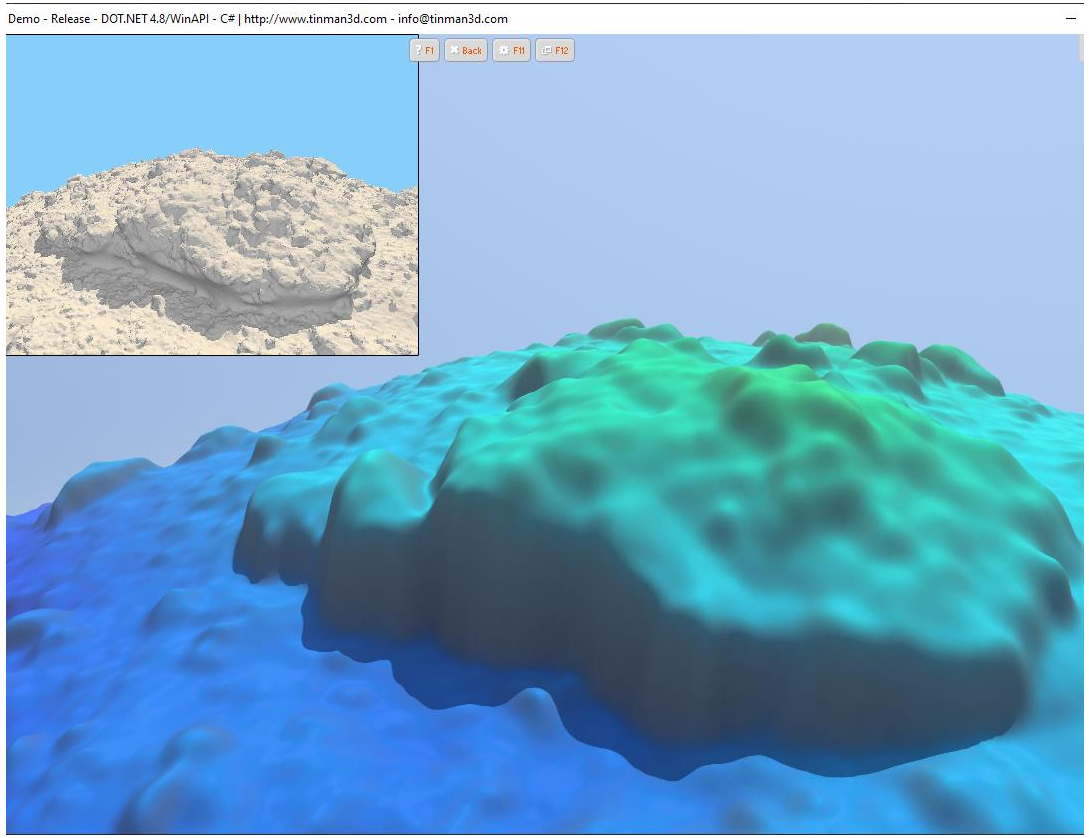
THE ELEVATION MAP



MAKING THE ELEVATION MAP #1

- Use a cubemap raster dataset
  - 6 x 8193 x 8193 is sufficient for Bennu
  - Interpolate to 65537 x 65537 for rendering
- Fit 3D model of Bennu into the cube
  - Put center-of-mass at the cube center
  - Use average distance from model vertex to cube center as “zero” elevation
- Cast rays from cube surface to center
- Compute elevation from ray hits
- **See demo source code for details!**

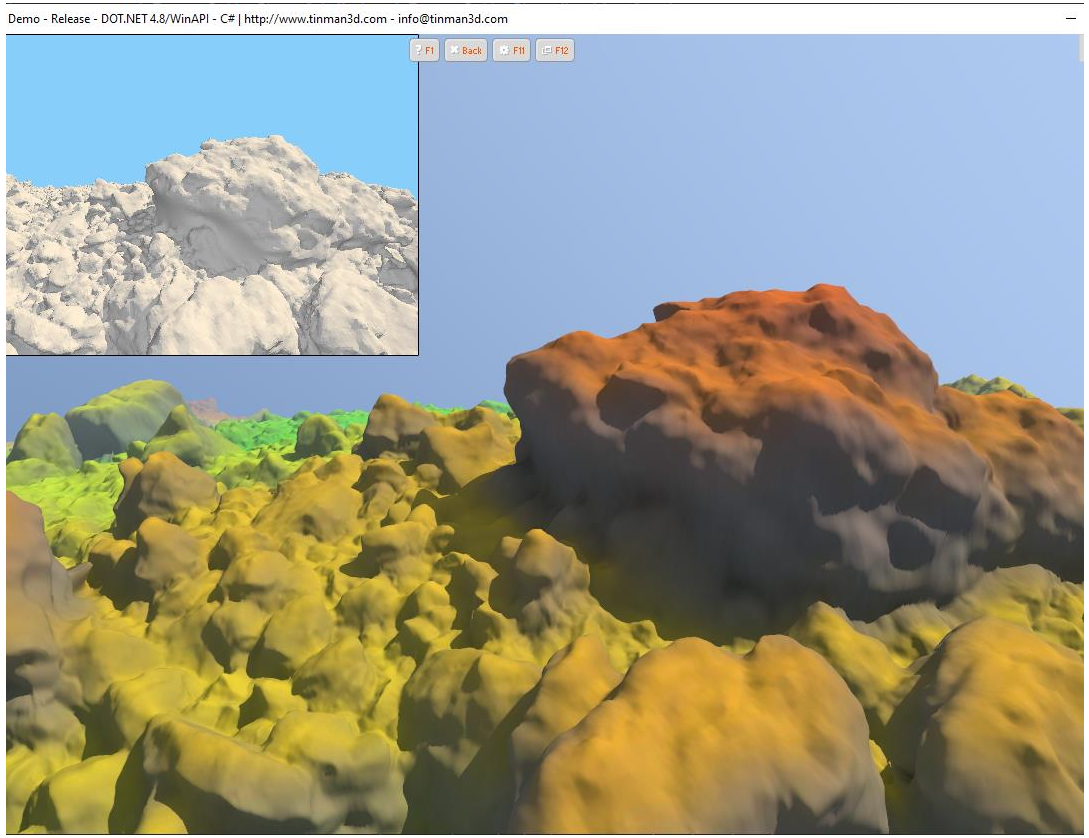
THE ELEVATION MAP



MAKING THE ELEVATION MAP #2

- Apply Gaussian filter to elevation map
  - A filter width of 60 samples is nice for Bennu
- This “base elevation map” is intentionally smooth…
  - which allows the “augmenting displacement map” to add non-2.5D terrain features, such as overhangs
- **Conflict of interest:**
  - Use displacement only where necessary, to break 2.5D logic as little as possible.
  - Capture as much non 2.5D features as possible, to increase visual appeal

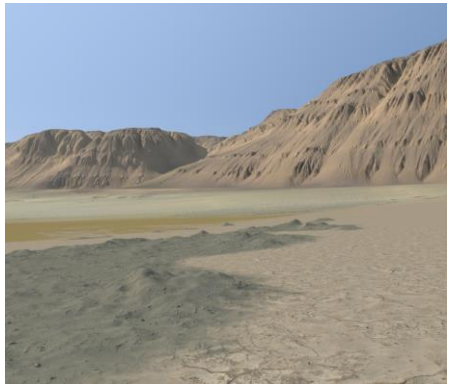
THE DISPLACEMENT MAP

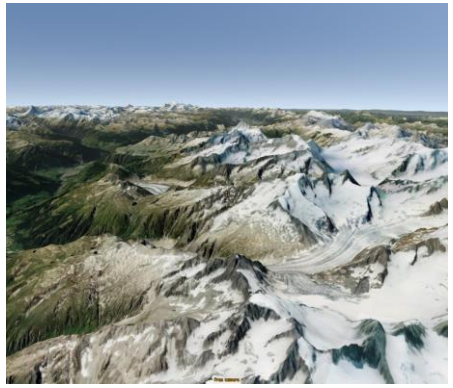


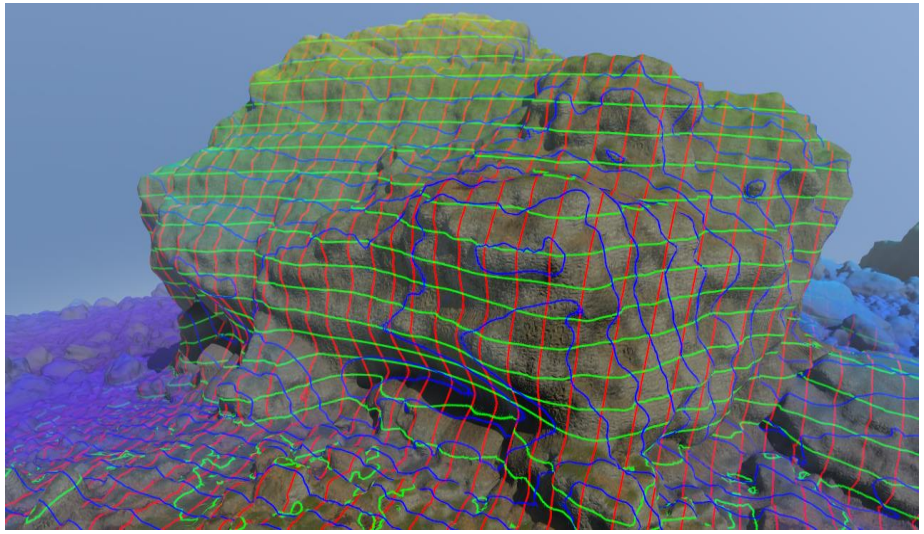
MAKING THE DISPLACEMENT MAP

- For each sample in the “base elevation model”:
  - Find nearest outward ray hit (along normal)
  - Find nearest inward ray hit (reverse normal)
  - Find nearest feature (omni-directional)
- Combine the three distances:
  - Use omni-directional distance to limit maximum displacement, to eliminate false attractors aka. “spikes”
  - Use spatial alignment of neighboring rays to limit maximum displacement, to reduce “wrinkles”
  - Choose smaller ray distance, to minimize ambiguity
- **See demo source code for details!**

CURVED TRI-PLANAR TEXTURE MAPPING







- Tri-planar texture mapping requires a fixed coordinate system:
  - Projection to the XY, XZ and YZ planes yields texture coordinates
  - Separate texturing passes are performed for each plane…
  - …and combined using the normal vector components as weights.
- **On a planetary body, there is no such fixed coordinate system!**
  - When moving, the “up-direction” changes continuously
  - Small problem for big bodies (Earth, Mars, Moon):
    near-surface views are indistinguishable from a flat terrain
  - Big problem for small bodies (Bennu):
    severe texture alignment resp. swimming artefacts are apparent

CURVED TRI-PLANAR TEXTURE MAPPING

SIGGRAPH 2021

PROBLEM: INTUITIVE UP-DIRECTION

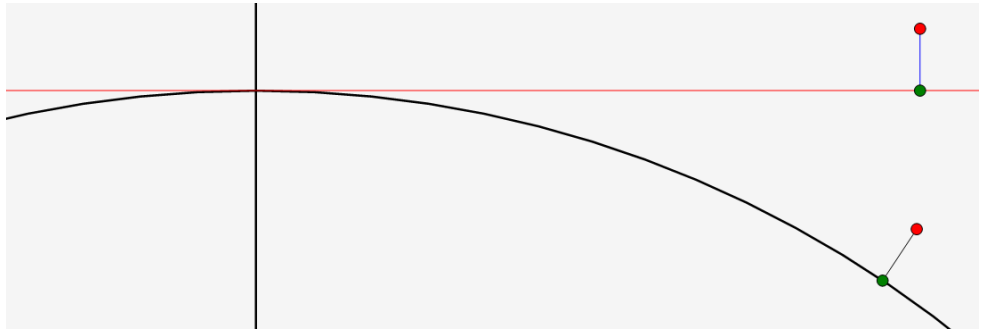
- Each location on the surface of a planetary body has its own intuitive “up-direction”
- The “up” here is “down” there
- **IDEA:** Define local coordinate system, centered at the camera, for example:
  - X : side, Y : up, Z : “north”
- Adjusting texture coordinate space to maintain the intuitive “up-direction” introduces massive texture swimming artefacts.

PROBLEM: TEXTURE SWIMMING #1

- **IDEA:** Apply a translation to texture coordinate space, to compensate for camera movement between frames.
- This reduces texture swimming, but not all of it
- The remaining swimming is now mostly along the “up-direction”…
- …which is caused by the “pitch” rotation that is inherent to the “up-direction” adjustment of texture coordinate space adjustment
- **See demo source code for details!**

PROBLEM: TEXTURE SWIMMING #2

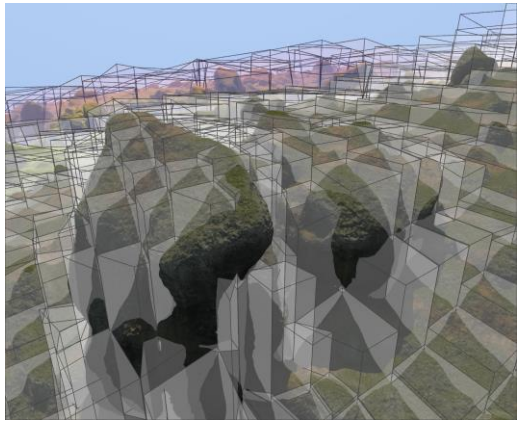
- **IDEA:** Apply a non-linear projection to texture coordinate space, to compensate for the “pitch” rotation.
- Scale XZ so that |XZ| is equal to the arc-length
- Replace Y with the height above the base sphere

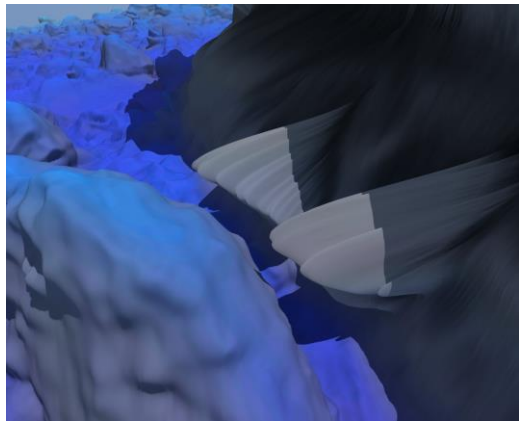


PROBLEM: TEXTURE SWIMMING #3

- Texture swimming has gone (almost)
- Still some swimming at distant terrain parts
- Not always noticeable, depends on camera position
- Caused by the “yaw” rotation which is inherent to the adjustment of texture coordinate space to align with “north”.
- **IDEA:** Apply a rotation around the Y-axis to texture coordinate space, to compensate for “north” alignment.
- **See demo source code for details!**

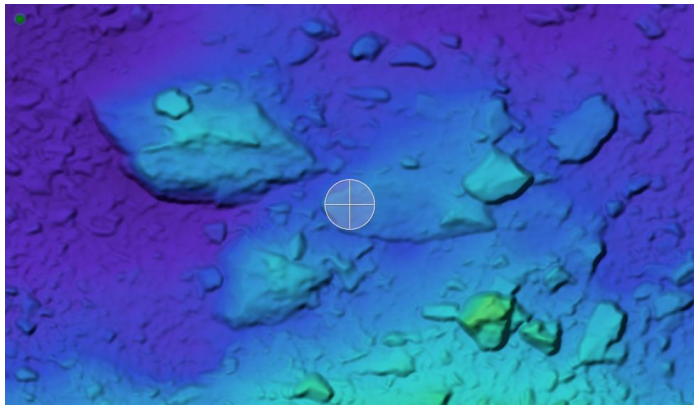
KNOWN PROBLEMS

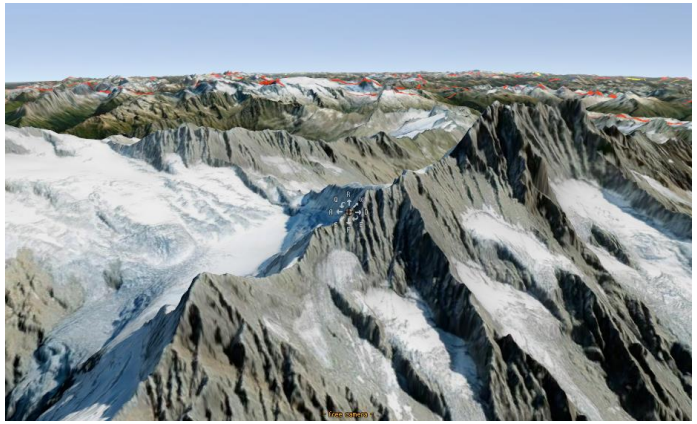




- Bounding boxes of 2.5D terrain quad-tree do not contain all terrain geometry when heavy displacement is present
- Horizon culling breaks with heavy displacement
- “Wrinkles” / self-intersection
  - may be reduced by limiting maximum ray distance
  - must blur elevation **more** to counteract
- “Macarons” / misinterpreted resp. inverted overhangs
  - must blur elevation **less** to counteract
  - https://en.wikipedia.org/wiki/Macaron
- “Spikes” / spatial ambiguity
  - ignore **front-facing outward ray hits** to counteract
  - ignore **back-facing inward ray hits** to counteract

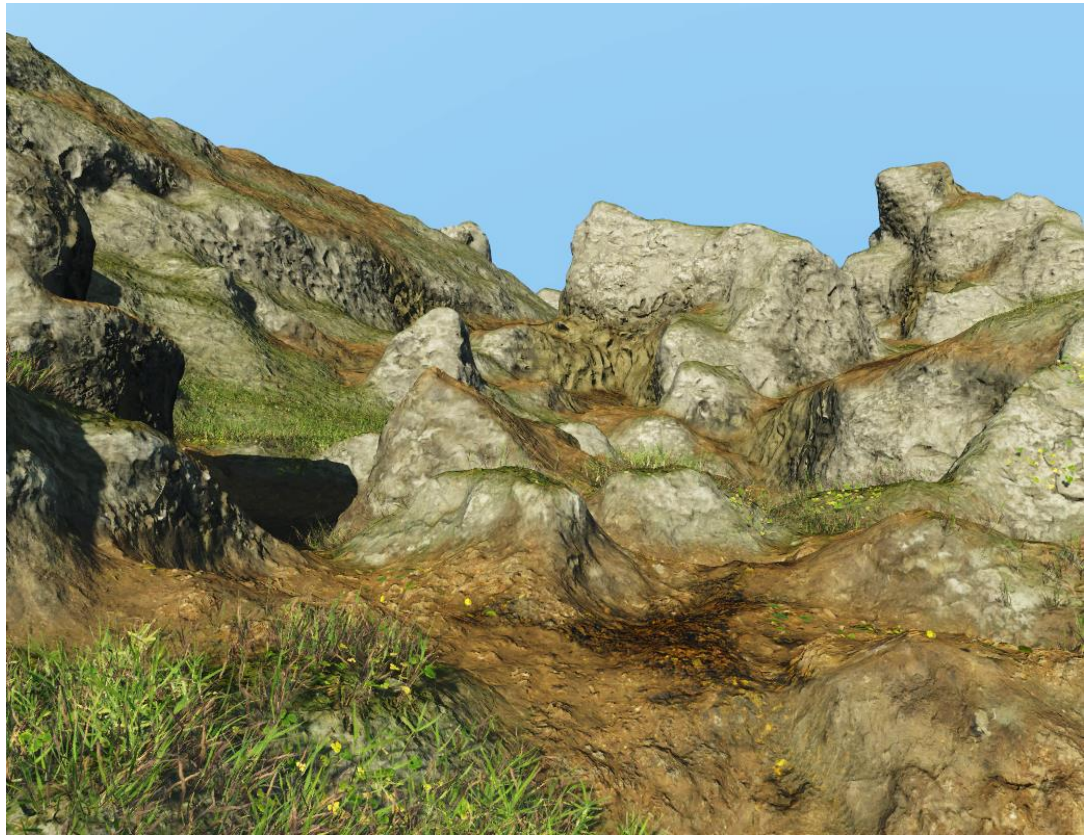
FUTURE WORK





- Use an **adaptive Gaussian filter kernel size**, to reduce displacement where possible and to improve displacement where necessary.
  - Based on mean elevation / slope / curvature and / or variance?
  - Based on coefficient map, provided by user?
  - Based on some iterative algorithm?
- Use **base ellipsoid** instead of base sphere, to better capture the overall form of Bennu (or any other planetary body).
  - Projection of texture coordinate space will be more complex.
  - What about precision and performance on the GPU?
  - Is it worth it?
- Apply technique to **other datasets**, for example: swissSURFACE3D https://www.swisstopo.admin.ch/en/geodata/height/surface3d.html

THE BENNU TERRAIN

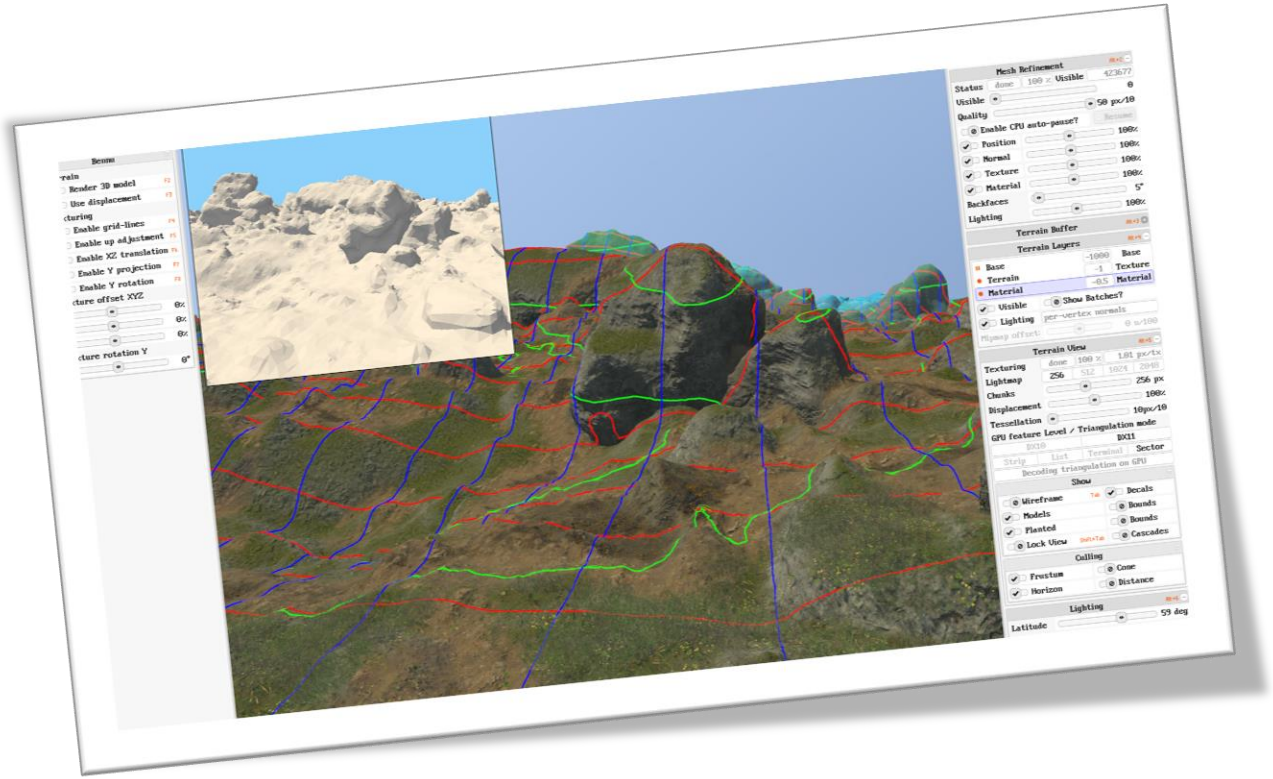


RECIPE FOR BENNU TERRAIN

- Load the high-resolution 3D model
- Scan the model with ray-casting
  - Build base elevation model
  - Build augmenting displacement model
- Use curved tri-planar texture mapping
  - Compute local-space for intuitive “up”
  - Fix swimming along XZ with translation
  - Fix swimming along Y with projection
  - Fix swimming around Y with rotation
- Use terrain engine to render heightmap + displacement

THANK YOU FOR WATCHING!

SIGGRAPH 2021



TRY FOR YOURSELF? GET THE DEMO!

- Go to this website: http://download.tinman3d.com/?dir=sdk/DEMO
- Download these:
  - SIGGRAPH.2021.zip
  - Tinman3D.Data.zip
  - Tinman3D.Demo.zip
- Extract ZIPs (*3zhfca87wov9*) and run the app:
  - “Tutorial_26_Bennu”
- Questions / problems? Write me@tinman3d.com!

MATTHIAS ENGLERT

FOUNDER

Fulda (Germany) - Tinman 3D - https://www.tinman3d.com/

Matthias Englert is a software enthusiast that grew up programming CGA, EGA and VGA boards - being fascinated by the possibilities of real-time computer graphics.

In 2012, Matthias created the Tinman3D SDK, a CPU-based 3D terrain-pipeline and rendering-engine, which uses modern GPUs for visual output.

Being thrilled by the advent of NVidia's Turing architecture, Matthias is now working on ways to integrate new GPU features into the SDK, for example Mesh Shaders and Ray Tracing.

To reach out for Matthias, just write him an email: me@tinman3d.com