Modeling Asteroid (101955) Bennu as a Real-time Terrain

Matthias Englert
Tinman 3D
Fulda, Germany
me@tinman3d.com

ABSTRACT

Using a commercial terrain engine [Englert 2012], we show how to model the complex surface of asteroid (101955) Bennu [Science 1999] with a hybrid approach that combines digital elevation models with static and dynamic displacement. Then we show how to adapt tri-planar material mapping (TRIMAP) to the curvature of planetary bodies, in order to effectively avoid all inherent texturing artefacts.

KEYWORDS

terrain rendering, continuous level-of-detail, displacement mapping, tri-planar mapping, map projection

ACM Reference Format:
Matthias Englert. 2021. Modeling Asteroid (101955) Bennu as a Real-time Terrain. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '21 Talks), August 09-13, 2021.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450623.3464628

1 INTRODUCTION

We describe a workflow that transforms a detailed 3D model of asteroid (101955) Bennu [Nolan et al. 2013] into a real-time continuous level-of-detail (CLOD) terrain [Englert 2020], applying standard rendering techniques to generate visuals that are suitable for edutainment and games.

The motivation is to be able to use conventional terrain techniques like height-maps [Lindstrom et al. 1996], displacement-maps [Karhu 2002] and tri-planar texturing [Weiss et al. 2020] to capture surfaces that are more complex and thus more interesting than conventional 2.5D terrains. Traditionally, those surface need to be processed with more elaborate techniques, for example view-dependent rendering of massive 3D models [Brüderlin et al. 2007] and 3D model sculpting [Peng et al. 2018].

Having been generated in the scope of the OSIRIS-REx mission [Nolan et al. 2013], the 3D model of Bennu captures fine-grained surface detail, such as holes and overhangs, which are difficult to represent using conventional terrain techniques. We show how to perform ray-casting over multiple stages, in order to obtain a base elevation model (BEM) and an augmenting displacement model (ADM), which can be fed into a standard terrain engine for real-time rendering.

Near-surface views exhibit the the bland look of the bare 3D model, so we show how to apply TRIMAP in order to make the surface look more appealing. We present details on the ray-casting stages, describing the algorithm for extracting the BEM and the ADM, as well as common data processing problems and viable solutions:

- Self-intersection aka. "wrinkles"
- Geometric ambiguity aka. "spikes"
- The "Macaron" effect

Finally, we show how to implement TRIMAP with adaptive displacement for planetary bodies, taking into account the curvature of the surface, in order to avoid texture swimming artefacts while maintaining an intuitive sense of the topocentric up-direction.

SIGGRAPH '21 Talks, August 09-13, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8373-8/21/08.
https://doi.org/10.1145/3450623.3464628

2 ATTENDEE TAKEAWAYS

- A working example in C# that demonstrates the algorithm that turns a detailed 3D model into a pair of elevation and displacement maps, which are suitable for consumption by real-time terrain rendering or conventional terrain content pipelines.
- Ready-to-use HLSL code that implements TRIMAP with adaptive displacement on planetary bodies, taking into account the curvature of the surface for small bodies as well as floating-point precision issues for large bodies.

3 3D MODEL IMPORT

The used 3D model of Bennu is an OBJ file of approximately 8.1 GB in size and contains over 177.000.000 triangular faces. Such data usually needs to be processed with dedicated tools before it may be consumed by rendering pipelines. Here we use Tinman 3D [Englert 2012] to process the input file into a 3D mesh that exhibits a spatial acceleration structure, which is essential for the subsequent model scan steps as well as for interactive Ground truth rendering.

4 3D MODEL SCAN

We use a variant of Ellipsoidal Cube Maps [Lambers and Kolb 2012] to define the BEM and the augmenting ADM. The 3D model is inscribed into the reference cube, then rays are cast to compute map samples. The maps have a resolution of 8193 by 8193 samples per cube map face, which produces acceptable results for small planetary bodies such as Bennu.

4.1 Cube Fitting

A simple approach is used to fit the 3D model into the reference cube: the cube center point is found by computing the center-of-mass of the triangles of the mesh; the semi-major and semi-minor radii of the base ellipsoid are computed from the distances of the mesh vertices to the center point.

4.2 Geocentric Scan

The first scan step computes the samples of the BEM: a ray is cast for each map sample, pointing into the opposite direction of the up-vector of the base ellipsoid and starting at a point that is well outside of the bounding sphere of the 3D model.

The resulting BEM exhibits discontinuities at places where the surface cannot be expressed with a conventional height-map (for example: caves, overhangs and holes), which renders it unsuitable for 2.5D terrain visualization at this point.

The BEM is blurred using a Gaussian filter kernel [Research 2016]. We treat the filter kernel size as a global parameter that may be modified by the user to tweak the result. The blurred BEM now approximates the surface of the 3D model and can already be used for terrain rendering, providing coarse detail only.

4.3 Topocentric Scan

A 3D terrain is created from the BEM, yielding vertex positions and normal-vectors for the map samples. These are used by the second scan step, which computes the samples of the ADM: two rays are cast, starting at the vertex position and being co-linear to the normal-vector. The displacement value is computed from the resulting distances and stored in the ADM.

5 TERRAIN RENDERING

Rendering the BEM and the displacement map as a real-time terrain is straight-forward. Here we use [Englert 2012], but other terrain engine capable of consuming elevation and displacement maps can be used as well.

5.1 Continuous Level-of-detail

Having an adaptive CLOD representation of the terrain ready on the CPU is beneficial for a variety of standard tasks (collision detection, physics simulation, planting). Here we use [Englert 2020] to render such a CPU representation efficiently on the GPU. Alternative approaches rely on the GPU [Weiss et al. 2020] and read back data from it, in order to be able to perform those CPU tasks.

5.2 Displacement Mapping

The displacement information in the ADM is incorporated into the CLOD representation of the terrain and influences vertex positions and normal-vectors. This happens outside of the scope of the GPU.

The terrain mesh, as seen by the GPU as input, has a smooth surface with per-vertex normal vectors. To increase visual detail, we apply adaptive GPU tessellation. The CLOD representation of the terrain contains per-vertex measures that can be used to compute near-optimal tessellation factors.

5.3 Tri-planar Mapping

TRIMAP is a well-known technique that is often applied to terrain rendering: world-space coordinates are used to derive texture coordinates, separate texturing steps are performed for the primary planes and the results are summed using the components of the respective normal-vector as weights.

Albeit conceptually easy, implementing TRIMAP might not necessarily be. If not done correctly, visual artefacts will appear, which might go unnoticed on quick looks but will look plain wrong on close looks. Typical examples are flipped axes in normal-map space and misaligned tangent-space. As reference, we provide a robust implementation including HLSL source code.

Traditional TRIMAP is not suited for use on a spherical body. The base plane of a traditional terrain has a constant up-vector, a spherical terrain does not; thus the orientation of the texture coordinate frame varies. To accommodate this, we present a two-step process.

First, a linear transformation is introduced that keeps texture coordinates in place while moving over the terrain. This solves the problem of mapping the sphere surface onto a plane, by allowing texture coordinate frames to shift in order to produce plausible visuals, instead of adhering to strict map projection properties.

Second, a spherical transformation is applied before TRIMAP, in order to eliminate texturing artefacts that are inherent for planetary bodies. For large planetary bodies such as Mars, these artefacts are not apparent because the curvature of the surface is visually indistinguishable from a plane (for near-surface views). For small planetary bodies like Bennu, these artefacts are severe and are visually not acceptable.

6 DEMONSTRATION

- Interactive 3D rendering of the full resolution 3D model (Ground truth)
- Real-time 3D rendering of the BEM and the ADM
- Visual examples of texturing artefacts and how to fix them
- Visual examples of data processing artefacts and how to minimize them

REFERENCES

B. Brüderlin, M. Heyer, and S. Pfützner. 2007. Interviews3D: A Platform for Interactive Handling of Massive Data Sets. *IEEE Computer Graphics and Applications* 27, 6 (2007), 48–59. https://doi.org/10.1109/MCG.2007.153

Matthias Englert. 2012. *Tinman 3D SDK - Real-time Terrain.* https://www.tinman3d.com

Matthias Englert. 2020. Using Mesh Shaders for Continuous Level-of-Detail Terrain Rendering. In *ACM SIGGRAPH 2020 Talks* (Virtual Event, USA) *(SIGGRAPH '20)*. Association for Computing Machinery, New York, NY, USA, Article 44, 2 pages. https://doi.org/10.1145/3388767.3407391

Kimmo Karhu. 2002. Displacement Mapping. (04 2002).

Martin Lambers and Andreas Kolb. 2012. Ellipsoidal Cube Maps for Accurate Rendering of Planetary-Scale Terrain Data. In *Pacific Graphics Short Papers*, Chris Bregler, Pedro Sander, and Michael Wimmer (Eds.). The Eurographics Association. https://doi.org/10.2312/PE/PG/PG2012short/005-010

Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust, and Gregory A. Turner. 1996. Real-Time, Continuous Level of Detail Rendering of Height Fields. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 109–118. https://doi.org/10.1145/237170.237217

Michael C. Nolan, Christopher Magri, Ellen S. Howell, Lance A.M. Benner, Jon D. Giorgini, Carl W. Hergenrother, R. Scott Hudson, Dante S. Lauretta, Jean-Luc Margot, Steven J. Ostro, and Daniel J. Scheeres. 2013. Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations. *Icarus* 226, 1 (2013), 629–640. https://doi.org/10.1016/j.icarus.2013.05.028

Mengqi Peng, Jun Xing, and Li-Yi Wei. 2018. Autocomplete 3D Sculpting. *ACM Trans. Graph.* 37, 4, Article 132 (July 2018), 15 pages. https://doi.org/10.1145/3197517.3201297

Wolfram Research. 2016. *GaussianFilter.* https://reference.wolfram.com/language/ref/GaussianFilter.html

NASA Science. 1999. *101955 Bennu.* https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/101955-bennu/overview/

Sebastian Weiss, Florian Bayer, and Rüdiger Westermann. 2020. Triplanar Displacement Mapping for Terrain Rendering. In *41st Annual Conference of the European Association for Computer Graphics, Eurographics 2020 - Short Papers, Norrköping, Sweden, May 25-29, 2020 [online only]*, Alexander Wilkie and Francesco Banterle (Eds.). Eurographics Association, 53–56. https://doi.org/10.2312/egs.20201016