Pacific Graphics (2012)
C. Bregler, P. Sander, and M. Wimmer (Editors)

Ellipsoidal Cube Maps for Accurate Rendering of Planetary-Scale Terrain Data

M. Lambers and A. Kolb

Computer Graphics Group
University of Siegen

Abstract

Advances in sensor technology lead to a rapidly growing number of terrain data sets with very high spatial resolution. To allow reliable visual analysis of this data, terrain data for planetary objects needs to be rendered with accurate reproduction of every detail.
This combination of very large scale and very fine detail is challenging for multiple reasons: the numerical accuracy of typical data types is not sufficient, simple spherical planet models fail to accurately represent the data, and distortions in map projections used for data storage lead to sampling problems.
In this paper, we propose the Ellipsoidal Cube Map model to address these problems. We demonstrate the possibilities of the model using a simple renderer implementation that achieves interactive frame rates for a variety of data sets for Earth, Moon, and Mars.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.3.8 [Computer Graphics]: Applications—

1. Introduction

Today, spaceborne and airborne Remote Sensing systems provide data sets for the Earth's surface with details on the decimeter or even centimeter scale. This is true both for image-like data, e.g. from multispectral sensors or Synthetic Aperture Radar (SAR) systems, and for elevation data, e.g. from Interferometric SAR (InSAR) or Light Detection And Ranging (LIDAR) systems. Additionally, high-resolution data sets increasingly become available for other objects in the solar system, such as the Moon and Mars.

This raises the problem of rendering high-resolution terrain data sets for planetary-scale objects with adequate precision to allow reliable visual analysis. Compared to terrain rendering systems that are only concerned with limited local areas, additional challenges in the areas of numerical accuracy, planet modelling, and terrain data map projection must be addressed.

In this paper, we propose the *Ellipsoidal Cube Map* (ECM) model. The main contributions are as follows:

- A data model that uses an ellipsoid as the common reference, to allow accurate interpretation of high-resolution data.
- A terrain data map projection that provides nearly uniform data sampling quality across the complete planet surface, without singularities at poles or decreasing quality in border regions.
- Methods to split geometry computations for rendering into two parts: static double-precision computations and dynamic single-precision computations. This allows accurate rendering using efficient single-precision rendering pipelines.

In contrast to previous approaches, the proposed techniques allow accurate rendering of planet-wide data sets at decimeter and centimeter resolutions across the complete planet surface. Furthermore, the model allows to generate or augment terrain data at rendering time for applications such as erosion simulations, procedural terrain detail generation, or interactive fusion of multiple data sets.

We demonstrate the techniques using a simple level-of-detail and rendering approach. Alternatively, the ECM model can be combined with existing performance-optimized terrain rendering approaches.

DOI: 10.2312/PE/PG/PG2012short/005-010

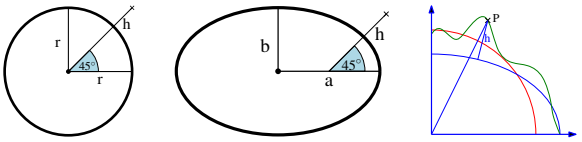

Figure 1: *Measurement of elevation h and latitude $\phi = 45°$ in a spherical planet model (left) and in an ellipsoidal planet model (middle). Right: transforming a point P on a planet surface (green) from an ellipsoidal planet model (blue) to a spherical planet model (red) requires exact knowledge of its elevation h; uncertainty in h results in uncertainty in the coordinates on the sphere surface.*

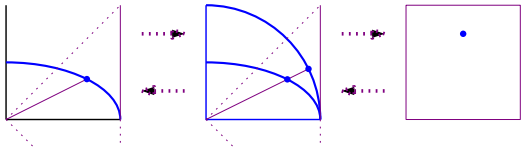

Figure 2: *Forward and inverse map projection. For forward projection, a point on the reference ellipsoid (left) is first shifted to the sphere (middle), and then projected onto the cube side (right).*

2. Related Work

Numerical Accuracy: Planetary-scale scenes require coordinates with large absolute values, and high-resolution terrain data requires small differences between these values to be preserved. The associated limitations of the single-precision floating point data type commonly used in rendering pipelines are well documented [Tho05, PG07, KLJ*09]. A recent discussion is given by Cozzi and Ring [CR11].

While using double-precision coordinate data for scene management is required, rendering pipelines typically use a single-precision data type for efficiency reasons. One approach to solve this problem is to divide the scene into multiple segments with local origins and render these segments sequentially with appropriate model-view matrices computed on the CPU in double-precision [LKR*97, RLIB99]. An alternative approach with simpler scene management is to shift the origin of the scene to the camera position for rendering [Tho05]. This preserves detail in vicinity to the camera, and limits precision loss to regions far away from the camera where it does not affect the rendering result.

Cignoni et al. use camera-centric base information for each patch of geometry and use linear interpolation in single-precision on the GPU across the patch, which introduces an error that needs to be accounted for [CGG*03]. Kooima et al. use specialized methods to refine spherical geometry in local coordinate systems on the GPU, but their texture coordinate precision is limited to meter scale on an Earth-sized planet [KLJ*09]. Lambers and Kolb use costly double-precision computations for selected intermediate steps on the GPU [LK09], leading to reduced rendering performance.

Planet Model: The sphere is the most widely used planet model [LKR*97, RLIB99, CGG*03, KLJ*09]. However, the true shape of most planetary objects is much better represented by a reference ellipsoid, which is an ellipsoid of rotation given by a semi-major axis a and a semi-minor axis b. The prevalent reference ellipsoid for the Earth is defined by the World Geodetic System 1984 (WGS84) standard [US 04]. Remote Sensing data sets are generally given relative to a reference ellipsoid for accuracy reasons.

A precise transformation of data sets to a spherical model requires exact knowledge of elevation information (see Fig. 1), but this transformation is complex, and elevation data with sufficient precision is often not available. None of the existing sphere-based approaches are documented to apply such a transformation. Instead, elevation data is usually interpreted along sphere normals. This introduces errors (see Fig. 7). To avoid the impractical transformation and all associated errors entirely, a rendering system must use the reference ellipsoid model. Only few approaches do this [WRH99, LK09].

Map Projection: A popular map projection is the equidistant cylindrical (or plate carrée) projection. Like all projections that map the complete planet surface, the plate carrée projection suffers from singularities, in this case at the poles. The associated sampling problems in the polar regions cause significant storage and data access overhead in the renderer as well as radial blur in the rendered image [KLJ*09].

The only way to avoid these problems is to use multiple maps. Kooima et al. combine plate carrée projection for the equatorial region and two additional projections for the polar regions using weighted averages for smooth transitions [KLJ*09]. Cignoni et al. inscribe a sphere into a cube and then use Gnomonic projection to map the sphere surface to the six cube sides [CGG*03]. See Fig. 3. Gnomonic projection suffers from shape and area distortions that rapidly increase away from the projection center [Sny87]. Lerbour et al. proposed an adjustment to Gnomonic projection [LMG10] that can reduce these distortions to some degree (see Sec. 3).

3. Ellipsoidal Cube Map

We propose the Ellipsoidal Cube Map (ECM), consisting of a reference ellipsoid, a circumscribing cube, and a map projection from the ellipsoid surface to the six cube sides.

The reasons for choosing a reference ellipsoid as the base model are given in Sec. 2. Since using a single map projection will always result in singularities [KLJ*09, Sny87], the base model must be subdivided. The circumscribing cube is a uniform subdivision that avoids special handling and blending of different regions, and allows simple and efficient quadtree-based data management and level of detail.

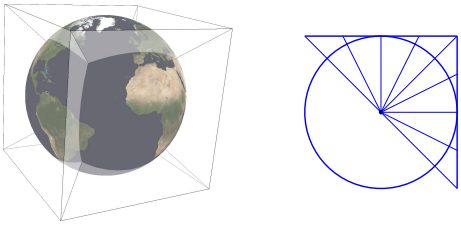

Figure 3: *Left: a spherical planet model inscribed into a cube. Right: Gnomonic projection maps the surface onto tangential cube sides.*

To be able to use the same map projection for each cube side, the ellipsoidal base model has to be shifted to a spherical model before applying the map projection, as shown in Fig. 2. Note that this shift applies to points directly on the reference surface, i.e. with zero elevation, and can therefore be performed without loss of precision.

With this base model shift, the remaining problem is to find a suitable projection from the sphere surface to the cube sides. There are basically two kinds of distortions to consider minimizing in a projection from the sphere surface to a plane: angle distortions and area distortions [FH05]. Angle distortions lead to shape distortions so that e.g. circular areas on a sphere surface are mapped to ellipsoidal areas on a plane. *Conformal* projections preserve angles. Area distortions cause equally sized areas on a sphere to be mapped to regions of different size on a plane. *Equal-area* projections preserve area ratios.

In a rendering system, both angle and area distortions cause sampling overhead and rendering artifacts [KLJ*09]. Since the sphere surface is not developable, a projection onto a plane cannot be both conformal and equal-area at the same time [Sny87, FH05]. For these reasons, one must choose a projection that limits both angle and area distortions to an acceptable degree.

A map projection that is applicable to the circumscribing cube must map each of the six sphere surface areas to square maps. We further require that each cube side is handled equally, which excludes the HEALPix projection that uses different projections for the polar sides and the equatorial sides of a cube [CR07].

We considered the following projections for use with the ECM model:

- Gnomonic projection (see Fig. 3). This straightforward method is used e.g. by Cignoni et al. [CGG*03]. It is neither conformal nor equal-area, and in fact distortions are known to increase rapidly with growing distance from the projection center [Sny87].
- Adjusted Gnomonic projection [LMG10]. The Gnomonic cube side coordinates $u, v \in [-1, 1]$ are transformed by $f(x) = \frac{4}{\pi} \cdot \operatorname{atan}(x)$ to reduce distortions.

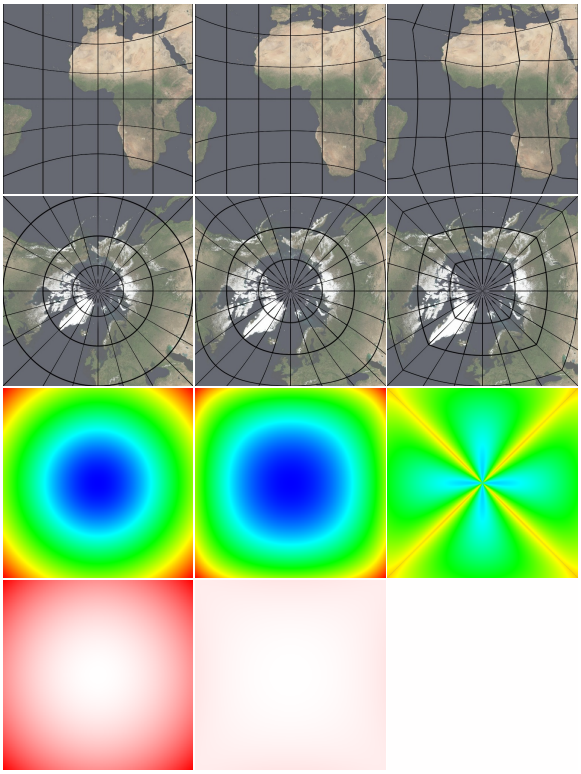

Figure 4: *Different projections from the sphere surface to a cube side. From left to right: Gnomonic projection, Adjusted Gnomonic projection, and Quadrilateralized Spherical Cube projection. The latitude and longitude lines in the first and second row have distances of 15°. The third row shows angular distortion color-coded from blue (no distortion) to red (maximum distortion). The fourth row shows area distortion color-coded from white (no distortion) to full red (maximum distortion).*

- Quadrilateralized Spherical Cube (QSC) projection. Chan and O'Neill [CO75] and O'Neill and Laubscher [OL76] developed a QSC model in which they inscribed a cube to a sphere and defined hierarchical structures on each cube side for data storage. For that purpose, they designed the QSC projection to be equal-area and at the same time limit angle/shape distortions.

For the visual comparison of the candidates shown in Fig. 4, we computed angle and area distortions across a cube side for each projection candidate based on the standard method of Tissot's Indicatrix as described by Snyder [Sny87]: angle distortion corresponds to Tissot's maximum angular deformation ω, and area distortion corresponds to differences in Tissot's areal scale factor s. The maximum angular distortion is 31.1° for both Gnomonic and Adjusted Gnomonic projection, and 24.9° for QSC projection. Area distortions are clearly strongest in Gnomonic projec-

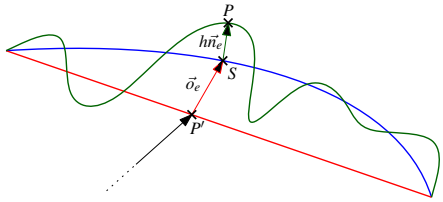

Figure 5: *Computation of camera-centric coordinates. P' is interpolated from the quad corners along the red line. The displacement vector $\vec{o}_e$ is the offset between P' and the corresponding position S on the ellipsoid surface (blue). The final position P on the true surface geometry (green) is computed by applying the elevation h along the normal $\vec{n}_e$.*

tion, with a maximum ratio of 5.19. Adjusted Gnomonic projection significantly reduces area distortions to a maximum ratio of 1.41. Per construction, QSC projection does not suffer from area distortions at all.

Based on these results, we chose the QSC projection for use with the ECM model.

Each cube side is the root of a quadtree hierarchy in which terrain data maps are stored. For a sphere with a given radius r, a quadtree level l provides a fixed, uniform resolution across the complete sphere surface, since the QSC projection is equal-area. For an ellipsoid with axes $a = r$ and $b < r$, small variations are introduced by the preceding shift to a sphere, depending on the ellipsoid flattening.

4. Geometry Computations for Rendering

As described in Sec. 2, scene management must use double precision, but for efficiency reasons it is desirable to use only single-precision in the rendering pipeline while still maintaining full accuracy, and that can be achieved using camera-centric coordinates for rendering.

The ECM model allows to split coordinate computations required for rendering one quad into static double-precision computations, performed once on the CPU, and dynamic single-precision computations, performed for each frame in the rendering pipeline.

Static double-precision computations:

- For each quad: cartesian planetocentric coordinates $Q_0, \ldots, Q_3$ of the quad corners on the ellipsoid surface.
- For each quad sample: the displacement vector $\vec{o}_e$ between a position P' bilinearly interpolated from $Q_0, \ldots, Q_3$ and the true ellipsoid surface position S, and the normal vector $\vec{n}_e$ at S. See Fig. 5.

$Q_0, \ldots, Q_3$ are stored in double-precision. To render with camera-centric coordinates, the double-precision camera coordinates C are subtracted, and the resulting camera-centric quad coordinates $Q'_0, \ldots, Q'_3$ are given to the rendering pipeline in single-precision.

The displacement and normal maps are computed in double-precision because they depend on planetary-scale coordinate values. The results, however, are local to the quad reference, and invariant to translation. Therefore, these maps are static and can be stored and used in single-precision. The symmetry of the ellipsoid can be exploited to reduce the number of computations.

Dynamic single-precision computations: To render a given quad, quad-relative vertex coordinates $(s,t) \in [0,1]^2$ need to be transformed to camera-centric cartesian coordinates $P \in \mathbb{R}^3$. Using the camera-centric quad corner coordinates $Q'_0, \ldots, Q'_3$, the precomputed displacement and normal maps, and the current elevation map, this can be done in single-precision in the rendering pipeline as follows:

- Compute a position P' by bilinearly interpolating $Q'_0, \ldots, Q'_3$ using (s,t).
- Read $\vec{o}_e$ from the displacement map, $\vec{n}_e$ from the normal map, and h from the elevation map at position (s,t).
- Compute the final position $P = P' + \vec{o}_e + h \cdot \vec{n}_e$.

With increasing quadtree levels, an ellipsoid surface patch represented by a quad converges to a plane, and thus displacement vectors converge to zero and normals converge to the plane normal. For this reason, if an application determines that the maximum length of a displacement vector for a given quadtree level is negligible, it can ignore displacement and normal maps starting from that level.

If a single static elevation data set is all that ever needs to be rendered, the static displacement map, normal map, and elevation map can be combined into a single static displacement map during preprocessing. Alternatively, keeping these maps separate allows to switch elevation data sets during rendering, and even to generate or augment elevation data on the fly, e.g. for terrain editing applications [ŠBBK08] or systems that interactively compose multimodal remote sensing data [LK09].

5. Implementation

Like other quadtree-based terrain rendering systems, our demonstration implementation consists of two parts: offline preprocessing and interactive rendering.

During preprocessing, quadtree hierarchies are built from the input elevation and texture data sets. An appropriate number of quadtree levels is chosen based on the nearly uniform resolution of each level. Preprocessing involves remapping the input data. In contrast to other approaches, precomputing geometry information is not necessary.

The interactive rendering step uses a common quadtree-based level of detail approach. In the first stage, the cube side quads that are necessary to render the current view are selected. In the second stage, the necessary data from the preprocessed terrain maps is asynchronously transferred to GPU memory (with caching) and rendered.

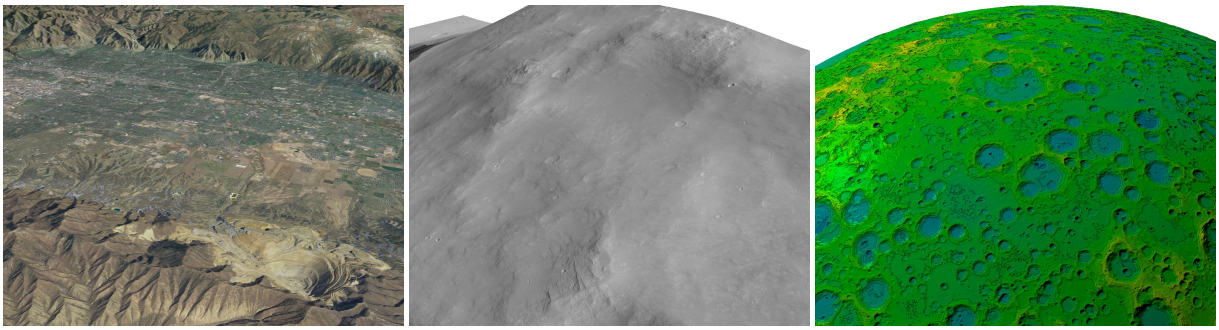

Figure 6: *Left: an area in Utah, with different elevation data sets fused at rendering time. Middle: a HiRISE image of Mars, combined with elevation data from the Mars Orbiter Laser Altimeter. Right: elevation data of the Moon, derived from Lunar Reconnaissance Orbiter Camera (LROC) images. The texture data is computed from the elevation data at rendering time.*

The first stage works as follows. The set of quads selected for rendering is initialized with the six root quads of the cube sides. The bounding box of each selected quad is projected to screen space, and unless it is outside the visible area or the number of pixels covered is less than $t = 1.5$ times the number of data samples it provides, it is replaced by its four subquads. This process is repeated until no more changes to the set of selected quads are necessary.

In the second stage, our demonstration implementation simply renders each quad with full geometric detail by rendering $2^k \times 2^k$ subquads for quads that provide $2^{k+1} \times 2^{k+1}$ data samples. This brute-force approach still results in interactive frame rates with current graphics hardware; alternatively, implementing a view-dependent level of detail mechanism is also possible. Skirts around each quad avoid discontinuities between quads of different quadtree levels [DSW09, LMG10].

6. Experimental Results

The experimental results described below were obtained on a Linux PC with an Intel Core i7-930 processor and an NVIDIA 480GTX graphics card. Preprocessing used 8 threads where possible. The quad size was 512×512 samples, and the viewport size was 1280×1024 pixels.

The Utah HRO 2006 data set provides photos with ca. 25 cm ground resolution. It consists of 3239 files, each storing 16000×16000 RGB samples using lossy compression. This corresponds to 2316.7 GiB of uncompressed input data. Preprocessing the data set, including lossless compression of the resulting quads, took 114 hours and 54 minutes.

The SRTM 90m Digital Elevation Database version 4 [JRNG08] provides data for Earth's land masses between $+60^\circ$ and -60° latitude. It consists of 872 files, each storing 6000×6000 16-bit samples (58.5 GiB uncompressed data). Preprocessing this data set took 9 hours and 30 minutes, including lossless compression.

Fig. 6 shows several example scenes. The left image shows a view of an area in Utah. The elevation data is combined from different data sets at 2 m and 5 m resolution at rendering time, allowing interactive examination and comparison of data sets. 21 quads from levels 6 – 11 are rendered at approximately 35 frames per second. The middle image shows a HiRISE image of Mars overlayed over elevation data from the Mars Orbiter Laser Altimeter. 21 quads from quadtree levels 10 – 13 are rendered at approximately 50 frames per second. The right image shows elevation data for the Moon derived from Lunar Reconnaissance Orbiter Camera images. Per-pixel lighting is applied. The color gradient texture, including isolines, is computed dynamically from the elevation data during rendering, allowing interactive examination of the elevation data. 25 quads from quadtree levels 2 – 3 are rendered at approximately 30 fps.

7. Discussion

The main characteristics of the ECM model (ellipsoidal model, QSC projection, accurate rendering in single-precision) affect both quality and efficiency.

The ellipsoidal model avoids errors caused by interpreting elevation data relative to a sphere. See Fig. 7. Approaches that use a sphere model should first transform data sets based on the best elevation model available, thereby reducing this error significantly for planetary objects with small flattening and high-quality elevation models such as Earth. However, this transformation is impractical and is not documented to be applied by any sphere-based approach. Our ellipsoid-based model avoids this transformation and associated errors entirely.

QSC projection avoids data access overhead and sampling problems caused by map projection distortions as described by Kooima et al. [KLJ*09]. It is more expensive than Gnomonic projection, but this additional cost only affects the preprocessing step. During rendering, only the computation of quad corner points $Q_0, \ldots, Q_3$ involves QSC projec-

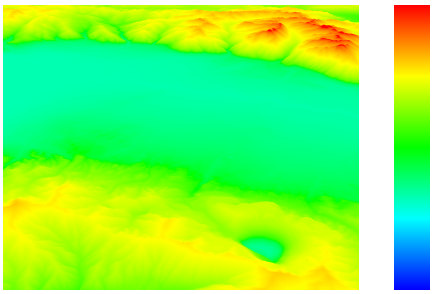

Figure 7: *The Utah scene (see Fig. 6 left) with color-coding of the error caused by interpreting elevations along a sphere normal instead of an ellipsoid normal. The used color range (right) encodes the error from 0 m (blue) to 11 m (red).*

tion. With typically 20 – 30 quads required for a view, these runtime costs are negligible.

The rendering approach used with the ECM model maintains full accuracy in single-precision and replaces costly and/or inaccurate geometry refinements, as used by previous approaches, with lookups of static precomputed data.

8. Conclusion

We propose the Ellipsoidal Cube Map model to overcome accuracy limitations of existing techniques to render planetary-scale terrain data sets. The ECM model is based on a reference ellipsoid circumscribed by a cube, and uses a map projection that preservers areas and angles better than alternatives. Accurate rendering can be achieved using only single-precision computations in the rendering pipeline. Generation or augmentation of elevation data at rendering time is possible.

The simple renderer described in Sec. 6 can demonstrate interactive frame rates at full detail level; alternatively, the ECM model can be combined with more sophisticated, performance-optimized terrain rendering methods.

Acknowledgements

We thank the providers of data used in this work: NASA Earth Observatory, the International Centre for Tropical Agriculture (CIAT), the Utah GIS portal, the Mars Orbiter Laser Altimeter (MOLA) and HiRISE missions, and the Lunar Reconnaissance Orbiter mission.

References

[CGG*03] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: Planet-sized batched dynamic adaptive meshes (P-BDAM). In *Proc. IEEE Visualization* (2003), pp. 147–154. 2, 3

[CO75] CHAN F., O'NEILL E.: *Feasibility Study of a Quadrilateralized Spherical Cube Earth Data Base*. Tech. Rep. EPRF 2-75 (CSC), Environmental Prediction Research Facility, Apr. 1975. 3

[CR07] CALABRETTA M. R., ROUKEMA B. F.: Mapping on the HEALPix grid. *Monthly Notices of the Royal Astronomical Society 381*, 2 (2007), 865–872. 3

[CR11] COZZI P., RING K.: *3D Engine Design for Virtual Globes*. CRC Press, June 2011. 2

[DSW09] DICK C., SCHNEIDER J., WESTERMANN R.: Efficient geometry compression for GPU-based decoding in realtime terrain rendering. *Computer Graphics Forum 28*, 1 (2009), 67–83. 5

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Dodgson N. A., Floater M. S., Sabin M. A., (Eds.), Mathematics and Visualization. Springer, 2005, pp. 157–186. 3

[JRNG08] JARVIS A., REUTER H., NELSON A., GUEVARA E.: Hole-filled seamless SRTM data v4, 2008. International Centre for Tropical Agriculture (CIAT). 5

[KLJ*09] KOOIMA R., LEIGH J., JOHNSON A., ROBERTS D., SUBBARAO M., DEFANTI T. A.: Planetary-scale terrain composition. *IEEE Trans. Visualization and Computer Graphics 15*, 5 (2009), 719–733. 2, 3, 5

[LK09] LAMBERS M., KOLB A.: GPU-based framework for distributed interactive 3D visualization of multimodal remote sensing data. In *Proc. IEEE Int. Geoscience and Remote Sensing Symposium* (2009), vol. 4, pp. IV–57–IV–60. 2, 4

[LKR*97] LINDSTROM P., KOLLER D., RIBARSKY W., HODGES L., DEN BOSCH A., FAUST N.: *An Integrated Global GIS and Visual Simulation System*. Tech. Rep. GIT-GVU-97-07, Georgia Institute of Technology, Mar. 1997. 2

[LMG10] LERBOUR R., MARVIE J.-E., GAUTRON P.: Adaptive real-time rendering of planetary terrains. In *Full Paper Proc. Int. Conf. Computer Graphics, Visualization and Computer Vision (WSCG)* (Feb. 2010). 2, 3, 5

[OL76] O'NEILL E., LAUBSCHER R.: *Extended Studies of a Quadrilateralized Spherical Cube Earth Data Base*. Tech. Rep. NEPRF 3-76 (CSC), Naval Environmental Prediction Research Facility, May 1976. 3

[PG07] PAJAROLA R., GOBBETTI E.: Survey of semi-regular multiresolution models for interactive terrain rendering. *Vis. Comput. 23*, 8 (2007), 583–605. 2

[RLIB99] REDDY M., LECLERC Y., IVERSON L., BLETTER N.: TerraVision II: Visualizing massive terrain databases in VRML. *IEEE Trans. Computer Graphics and Applications 19*, 2 (Mar. 1999), 30–38. 2

[ŠBBK08] ŠT'AVA O., BENEŠ B., BRISBIN M., KŘIVÁNEK J.: Interactive terrain modeling using hydraulic erosion. In *Proc. Eurographics Symposium on Computer Animation* (2008), pp. 201–210. 4

[Sny87] SNYDER J.: *Map projections—a working manual*, vol. 1395 of *Professional Paper*. US Geological Survey, 1987. 2, 3

[Tho05] THORNE C.: Using a floating origin to improve fidelity and performance of large, distributed virtual worlds. In *Proc. Int. Conf. on Cyberworlds* (2005), pp. 263–270. 2

[US 04] US NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY: World Geodetic System 1984 (WGS84). http://earth-info.nga.mil/GandG/wgs84/, 2004. Accessed 2012-04-30. 2

[WRH99] WARTELL Z., RIBARSKY W., HODGES L.: *Efficient Ray Intersection for Visualization and Navigation of Global Terrain using Spheroidal Height-Augmented Quadtrees*. Tech. Rep. GIT-GVU-99-20, Georgia Institute of Technology, 1999. 2