

Survey of Cube Mapping Methods in Interactive Computer Graphics

M. Lambers

Received: date / Accepted: date

Abstract The standard cube mapping technique implemented in graphics pipelines, while useful in many scenarios, has significant shortcomings for important application areas in interactive Computer Graphics, e.g. dynamic environment mapping, omnidirectional shadow maps, or planetary-scale terrain rendering. Many alternative mapping methods have been proposed over the years with the purpose of reducing area and/or angular distortions. In this paper, we give an overview of methods suitable for interactive applications, and analyze their properties. Furthermore, we evaluate a set of additional transformation functions and identify a simple new method with favorable distortion properties.

Keywords Cube Maps · Environment Maps

1 Introduction

Mappings between a sphere surface and a planar surface have been studied for centuries, starting with world maps and celestial maps [26]. Since the sphere surface is not developable, such mappings always exhibit either area or angular distortions, and often both. Equal-area mappings preserve area ratios, at the cost of large angular distortions, and conformal mappings preserve angles locally, at the cost of large area distortions.

This is a post-peer-review, pre-copyedit version of an article published in The Visual Computer. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s00371-019-01708-4>

M. Lambers
Computer Graphics Group
University of Siegen
D-57076 Siegen
Tel.: +49-271-7402842
E-mail: martin.lambers@uni-siegen.de

Mapping the complete sphere surface to a circle or square results in strong distortions [15]. Mapping portions of the sphere surface to corresponding faces of a polyhedron significantly lowers these distortions as the number of faces grows [27], at the cost of interruptions at the face boundaries that often manifest themselves as C^1 discontinuities of the mapping function.

In Computer Graphics, especially in interactive applications, using a cube as the polyhedron is of particular interest, since the number of faces is low and each face is a square, which eases the management of image-like data as well as the application of hierarchical methods such as quad trees or mipmaps. The cube map method implemented by standard graphics pipelines is the simplest form of mapping between cube and sphere surface. It is equivalent to Gnomonic projection for each cube face, and exhibits strong area and angular distortions.

Application areas for mappings between cube and sphere surfaces in interactive Computer Graphics include dynamic environmental maps for illumination [12] and shadow computations [25], planetary-scale terrain rendering [6], and procedural texturing [32]. Though these areas seem diverse, they all map image-like data that is available on a sphere surface to a cube surface, and during rendering sample that cube surface to reconstruct the original sphere surface data. In the case of environment mapping, the data on the sphere surface is given by infinite views in all directions from a single point. In the case of planetary rendering or procedural texturing, the data is directly associated with a sphere surface. In both cases, the methods for forward mapping (sphere to cube) and inverse mapping (cube to sphere) are the same.

In all the aforementioned application areas, alternatives to the standard cube map have been proposed to

avoid its distortion problems: with reduced map distortions, lower cube map resolutions can be used to achieve a comparable sampling quality during rendering, resulting in lower memory consumption and computational costs.

This paper gives an overview of the methods that have been proposed, with a focus on applications in interactive Computer Graphics. Formulas for forward and inverse mapping are given, and the methods are analyzed and compared regarding key properties. Furthermore, a new method is identified in a set of candidate methods inspired by previous approaches. This new method combines simplicity with favorable distortion properties.

Sec. 3 details the needs of example application areas and resulting requirements for cube mapping methods. Sec. 4 categorizes the known mapping methods and provides implementation details. Sec. 5 provides numerical analysis results for relevant methods. Sec. 6 concludes with recommendations. The supplementary material contains C++ source code that implements all mapping and analysis methods used in this paper.

2 Related Work

In this paper we focus on mapping methods for use in interactive Computer Graphics applications, specifically those that sample cube maps for rendering purposes. This requires that the mapping function is continuous along all cube edges, which excludes e.g. the equal-area mapping method proposed by Arvo [1] since it exhibits discontinuities along several cube edges [32]. Furthermore, the subdivision of the sphere into six areas that are subsequently mapped to cube faces must match the standard cube map subdivision so that common graphics pipeline functionality can be leveraged to access and filter cube maps. This excludes the HEALPix cube map scheme [4] and variants [31], and the Isocube map [30], which use six areas of equal size but differing shape (polar versus equatorial areas).

Furthermore, we exclude methods that have prohibitively high computational costs or require iterative approximations. This exclusion especially affects mapping methods that are *exactly* equal-area or conformal: equal-area mappings require iterative approximations [27, 11] or complex and costly mathematical expressions [24] in either the forward or inverse mapping (the QSC method described in Sec. 4.2, despite its complexity, is the cheapest equal-area method and is included here because it has been used in planetary terrain rendering), and conformal mappings are based on Jacobian elliptic functions [18] or Taylor series approximations thereof [22]. Other affected mappings in-

clude the method used in the Outerra planetary 3D engine [13] which requires iterative computations [6], the original COBE method [5] which suffers from inversion precision problems [3, 6], and the polynomial and COBE-variant methods proposed by Zucker and Higashi which only achieve acceptable inversion precision using iterative refinement methods [32].

The methods presented in this paper provide trade-offs between area and angular distortions, while keeping complexity and computational costs low.

Zucker and Higashi compared a few of the methods described in this paper with respect to area distortions and computational costs, but did not consider angular distortions [32]. We consider angular distortions, too, because their impact on sampling quality is not negligible [28, 14, 12, 6]. Furthermore, while Zucker and Higashi aimed to minimize the area distortion RSME, we aim to minimize the maximum distortion errors, i.e. to optimize the worst case behaviour.

A different small subset of the methods described in this paper was compared by Lambers and Kolb [16] and Dimitrijević et al. [6], but with a strong focus on the needs of planetary-scale terrain rendering. Consequently, Dimitrijević et al. apply quality measures that were derived from a method to use textures in this context. In contrast, we address a more general application field and therefore use more general quality measurement methods based on the standard analysis of Tissot's Indicatrix [26].

We include additional methods from the field of environment mapping into our analysis (Continuous-Cube [10] and UniCube [12]), and we systematically evaluate a set of new methods inspired by previous approaches, which have favorable distortion properties while being simple to implement and cheap to compute.

3 Application Area Requirements

Gathering of information from a cube map works similarly for all application areas. Sampling a standard cube map is typically done by specifying a lookup vector \mathbf{d} . From this vector, the cube face and the coordinates on that face are easily determined. Transforming a lookup vector \mathbf{d} from standard cube map space to one of the alternative cube map spaces described in Sec. 4 requires the forward transformation function f .

Application requirements differ in the cube map creation step. For applications in pseudorandom sample point distribution [32] or in planetary terrain rendering where the cube map is created by sampling image-like data given in some other map space [16], the inverse transformation f^{-1} is needed: for each sample point in

the chosen cube map space, first the standard cube map space coordinates are computed using f^{-1} , and these are then transformed to the input map space to sample the original data. In this case, it is important that the inverse of f is numerically precise to avoid sampling at wrong positions during the rendering stage. This is why we excluded methods relying on iterative approximations.

If, on the other hand, the cube map is created by rendering geometry into the cube faces, as in dynamic environment mapping [12] or for omnidirectional shadow maps, then only the forward transformation f is required. When rendering into a cube face, the x and y components of the normalized device coordinates are equivalent to the standard cube map coordinates for the current cube face, and accordingly can be transformed to an alternative cube map space using the forward transformation f in a vertex shader.

The problem with this approach is that straight lines do not map to straight lines anymore, as if one would render onto a curved surface instead of a plane [8]. Similar problems arise with parabolic maps for shadow mapping [25] or with lens distortion rendering via geometry preprocessing [17]. Since the error that is introduced grows with the screen space line length, finely tessellated geometry in the scene is required to keep that error small. Another problem, related to the first one, is that straight lines that cross cube faces may appear to be discontinuous. The remedy proposed by Ho et al. for their Unicube method [12] can be applied to any of the mapping methods listed in this paper.

In interactive Computer Graphics applications, the function f is typically implemented in a shader and should therefore be cheap to compute. To achieve good sampling quality, area distortions should be low. Angular distortions should not be too high either since they also affect sampling quality [28, 14, 12, 6]. It therefore makes sense to choose a method with low maximum area distortion error and acceptable angular distortions to guarantee a minimum sampling quality.

4 Cube Map Methods

In this section, we describe all methods in detail, starting with the standard cube map as implemented in graphics pipelines. Subsequent methods are listed in chronological order.

Analogous to Zucker and Higashi [32], we formulate all mapping methods as modifications of the standard cube map approach. This allows to leverage the hardware accelerated graphics pipeline functionality to get cube map coordinates that then only need to be adjusted.

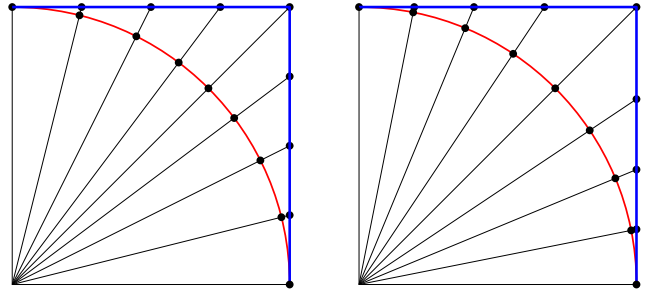


Fig. 1 Gnomonic projection (left) use even sample distances on the cube (blue), resulting in uneven sample distances on the sphere (red). With an adjustment function (right), sample distances are more evenly distributed on the sphere.

4.1 Standard Cube Map

The standard cube map implemented in most graphics pipelines was proposed by Greene [9]. It provides a mapping between the unit sphere and the unit cube surfaces by applying Gnomonic projection centered on each cube face, i.e. on the cartesian coordinate axes: sphere surface coordinates (x, y, z) are to mapped to coordinates $(u, v) \in [-1, +1]^2$ on cube face $i \in 0, \dots, 5$, with i corresponding to $+x, -x, +y, -y, +z, -z$.

A great advantage of the standard cube map is that cube maps can be easily created in standard graphics pipelines by rendering one image for each cube side. This is equivalent of shooting rays from the origin to form equidistant grids on the cube faces, and mapping the intersections with the cube and sphere surfaces to each other. See the left side of Fig. 1.

Gnomonic projection is known to suffer from area and angular distortions that grow rapidly with increasing distance from the projection center [26] (in our case, the centers of each cube face). The alternative mappings described in the remainder of this section all strive to reduce these distortions. However, as a consequence, the creation of cube maps by rendering images becomes more complex; see Sec. 3 for details.

We describe each method in terms of modifying the standard cube map coordinates (u, v) on one cube face to arrive at new coordinates (u', v') on the same cube face, either using a univariate function $f : [-1, 1] \rightarrow [-1, 1]$ applied equally to both u and v , or using a bivariate function $f : [-1, 1]^2 \rightarrow [-1, 1]^2$ applied to (u, v) .

In practice, univariate functions f can be applied to all components of the vector $\frac{\mathbf{d}}{\max(|d_x|, |d_y|, |d_z|)}$, where \mathbf{d} is the cube map lookup direction. This works since $f(-1) = -1$ and $f(1) = 1$, so only the cube face coordinates are actually modified [12].

4.2 Quadrilateralized Spherical Cube

O'Neill and Laubscher [21] developed the Quadrilateralized Spherical Cube (QSC) model based on previous work by Chan and O'Neill [5]. 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 angular distortions. In Computer Graphics, QSC has been used in planetary terrain rendering [16]. Although its computation is comparably complex and costly, we include this method here since it is the only known equal-area method with an analytical inverse.

$$\begin{aligned}
 f\left(\begin{matrix} u \\ v \end{matrix}\right) &= \tan(\nu) \begin{pmatrix} \cos(\mu) \\ \sin(\mu) \end{pmatrix}, \text{ with} \\
 \varphi &= \cos^{-1}\left(\frac{1}{\sqrt{u^2 + v^2 + 1}}\right) \\
 \theta &= \text{atan2}(u, -v) \\
 \mu &= \tan^{-1}\left(\frac{12}{\pi}\right) \\
 &\quad \left(\theta + \cos^{-1}\left(\sin(\theta) \cos\left(\frac{\pi}{4}\right)\right) - \frac{\pi}{2}\right) \\
 t &= 1 - \cos\left(\tan^{-1}\left(\frac{1}{\cos(\theta)}\right)\right) \\
 \tan(\nu) &= \sqrt{\frac{1 - \cos(\varphi)}{\cos^2(\mu)t}} \\
 f^{-1}\left(\begin{matrix} u' \\ v' \end{matrix}\right) &= \tan(\varphi) \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix}, \text{ with} \\
 \tan(\nu) &= \sqrt{u'^2 + v'^2} \\
 \mu &= \text{atan2}(v', u') \\
 \theta &= \tan^{-1}\left(\frac{\sin\left(\frac{\pi}{12} \tan(\mu)\right)}{\cos\left(\frac{\pi}{12} \tan(\mu)\right) - \frac{1}{\sqrt{2}}}\right) \\
 t &= 1 - \cos\left(\tan^{-1}\left(\frac{1}{\cos(\theta)}\right)\right) \\
 \varphi &= \cos^{-1}(1 - \cos^2(\mu) \tan^2(\nu)t)
 \end{aligned} \tag{1}$$

The above formulae assume $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, i.e. they apply only to one quarter of a cube face. The other quarters are handled by rotating them into the quarter of definition. This introduces \mathcal{C}^1 discontinuities at the cube face diagonals. While this method exhibits no area distortions, the average angular distortions of this equal-area method are quite strong. However, the maximum angular distortion error is lower than for all other methods.

4.3 Tangent Adjustment

The tangent adjustment of the standard cube map modifies the cube map coordinates in the following way:

$$\begin{aligned}
 f(w) &= \frac{\pi}{4} \tan^{-1}(w) \\
 f^{-1}(w') &= \tan\left(w' \frac{\pi}{4}\right)
 \end{aligned} \tag{2}$$

This method has been rediscovered repeatedly. In the context of Computer Graphics, it was used by Lerbou et al. for planetary terrain rendering [19] and by Bitterli et al. in dynamic environment mapping [2]. Earlier uses of the method can be found in other fields [22, 23].

The motivation for this method is that points on the sphere are distributed more evenly, thereby reducing distortions. See Fig. 1.

Zucker and Higashi introduce a parameter c [32]:

$$\begin{aligned}
 f(w) &= \frac{\tan^{-1}(cw)}{\tan^{-1}(c)} \\
 f^{-1}(w') &= \frac{\tan(w' \tan^{-1}(c))}{c}
 \end{aligned}$$

They numerically find $c \approx 1.1823$ to be optimal in terms of area distortion RMSE reduction, while Eq. 2 uses $c = 1$.

4.4 Nowell's Method

Nowell devised a mapping from cube surface to sphere surface based on cartesian coordinates [20, 29]. The formulas below assume that a given point (x, y, z) on the sphere surface maps to (u, v) on the cube face at $x = 1$; the other faces are handled by symmetry.

$$\begin{aligned}
 f(x, y, z) &= \begin{pmatrix} \frac{\text{sgn}(y)}{\sqrt{2}} \sqrt{t + 2y^2 - 2z^2 + 3} \\ \frac{\text{sgn}(z)}{\sqrt{2}} \sqrt{t - 2y^2 + 2z^2 + 3} \end{pmatrix}, \text{ with} \\
 t &= -\sqrt{(2z^2 - 2y^2 - 3)^2 - 24y^2} \\
 f^{-1}(u', v') &= \begin{pmatrix} \sqrt{1 - \frac{u'^2}{2} - \frac{v'^2}{2} + \frac{u'^2 v'^2}{3}} \\ u' \sqrt{\frac{1 - v'^2}{2} + \frac{v'^2}{3}} \\ v' \sqrt{\frac{1 - u'^2}{2} + \frac{u'^2}{3}} \end{pmatrix}
 \end{aligned} \tag{3}$$

This method has been used in environment mapping [29]. Its area and angular distortions are a clear improvement over standard cube maps, but better methods exist. Furthermore, the formulation in terms of cartesian coordinates complicates its application in a graphics pipeline because the hardware support for cube mapping cannot be used (or has to be undone).

4.5 Continuous Cube

The Continuous Cube method was introduced by Grimm and Niebruegge for the purpose of environment mapping [10]. The original formulation is based on cartesian coordinates. Here we give the formulation based on cube face coordinates so that the standard cube map functionality can be used:

$$\begin{aligned} f\left(\begin{matrix} u \\ v \end{matrix}\right) &= \left(\begin{matrix} \frac{\tan^{-1}\left(\frac{u}{\sqrt{2}}\right)}{\sin^{-1}\left(\frac{1}{\sqrt{3}}\right)} \\ \frac{\tan^{-1}(v \cdot \cos(\tan^{-1}(u)))}{\sin^{-1}\left(\frac{1}{\sqrt{2+u^2}}\right)} \end{matrix} \right) \\ f^{-1}\left(\begin{matrix} u' \\ v' \end{matrix}\right) &= \left(\begin{matrix} t \\ \frac{\tan\left(v' \sin^{-1}\left(\frac{1}{\sqrt{2+t^2}}\right)\right)}{\cos(\tan^{-1}(t))} \end{matrix} \right), \text{ with} \\ t &= \sqrt{2} \tan\left(u' \sin^{-1}\left(\frac{1}{\sqrt{3}}\right)\right) \end{aligned} \quad (4)$$

Similar to the Nowell method, the area and angular distortions of the Continuous Cube are an improvement over standard cube maps.

4.6 Unicube

The Unicube map was proposed by Ho et al. [12] as an improvement over the previous Isocube map [30]. The following adjustment function was derived to improve sampling uniformity compared to both the standard cube map and the Isocube map:

$$\begin{aligned} f(w) &= \frac{6}{\pi} \sin^{-1}\left(\frac{w}{\sqrt{2w^2 + 2}}\right) \\ f^{-1}(w') &= \frac{\sin\left(\frac{\pi}{6}w'\right)}{\sqrt{\frac{1}{2} - \sin^2\left(\frac{\pi}{6}w'\right)}} \end{aligned} \quad (5)$$

The Unicube method has low area distortions.

4.7 Everitt's Method

Zucker and Higashi [32] extracted the following adjustment function from source code provided by Everitt [7]:

$$\begin{aligned} f(w) &= w(c + (1 - c)|w|) \\ f^{-1}(w') &= \text{sgn}(w') \left(\frac{c - \sqrt{c^2 - 4(c - 1)|w'|}}{2(c - 1)} \right) \end{aligned} \quad (6)$$

While Everitt originally used $c = 1.5$ and $c = 1.375$, Zucker and Higashi determined numerically that $c \approx 1.4511$ is optimal in terms of area distortion RMSE minimization.

4.8 Sigmoid Adjustment (New Method)

The aforementioned adjustment functions and their inversions all take the shape of a sigmoid function $[-1, 1] \rightarrow [-1, 1]$. In order to identify adjustment function candidates with favorable properties, we took different analytically invertible sigmoid functions and introduced a parameter c to each of them to tune their behaviour. In particular, we tested algebraic sigmoid functions

$$f(w) = \frac{w\sqrt{1+c^2}}{\sqrt{1+c^2w^2}},$$

logistic sigmoid functions

$$f(w) = \frac{2e^c + 2}{e^c - 1} \left(\frac{1}{1 + e^{-cw}} - \frac{1}{2} \right),$$

smoothstep functions

$$f(w) = \frac{\frac{1}{2}(3(cw+1)^2 - (cw+1)^3) - 1}{\frac{1}{2}(3(c+1)^2 - (c+1)^3) - 1}, \quad c \in (0, 1],$$

the hyperbolic tangent function

$$f(w) = \frac{\tanh(cw)}{\tanh(c)},$$

and the parameterized tangent function described in Sec. 4.3.

Varying the parameter c can aim to minimize either the maximum error or the RMSE of either area or angular distortions. The resulting values for c will differ; see the first two entries in Tab. 2 for an example.

We minimize the maximum area distortion error, thus improving the worst case behaviour of each mapping method. Note that all sigmoid function variations have nearly the same maximum angular distortion error (reached at the cube face borders, where the difference to the standard cube map method reaches its minimum), except for the smoothstep function, which is worse. Therefore, trying to minimize this error does not make sense. Note also that while the area and angular RMSEs are not minimal when minimizing the maximum area distortion error, they are still in an acceptable range.

As a result of our tests, we recommend the algebraic sigmoid function with parameter $c = 0.8700$. See Sec. 5 for details. This function has the additional advantage of being cheap to compute:

$$\begin{aligned} f(w) &= w\sqrt{\frac{1+c^2}{1+c^2w^2}} \\ f^{-1}(w') &= \frac{w'}{\sqrt{1+c^2-c^2w'^2}} \end{aligned} \quad (7)$$

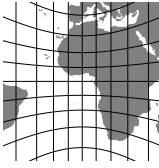
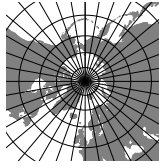


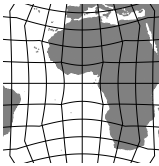
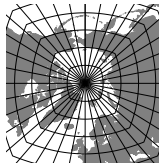

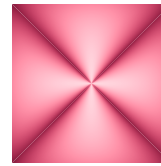
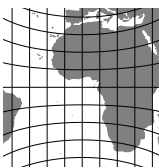
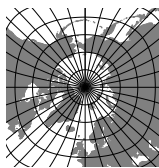
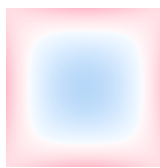

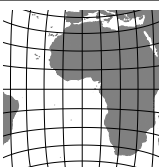
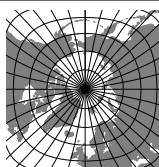


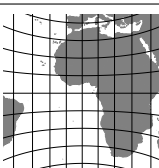
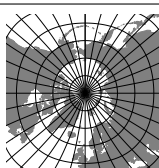
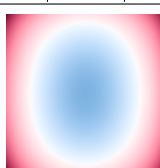
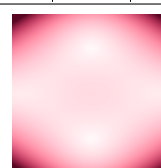
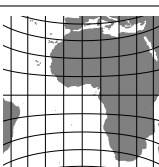
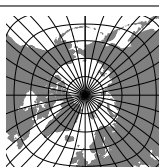
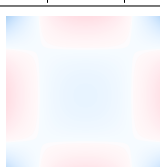
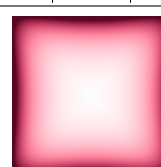
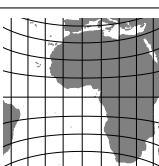
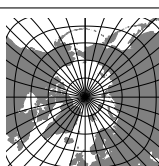
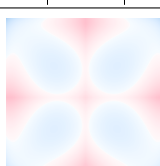
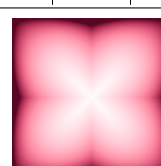
Method	Example cube face at $x=1$	Example cube face at $z=1$	D_A (ideal: 1) avg max err RMSE	D_I (ideal: 1) avg max err RMSE	Remarks
Standard cube map			 1.153 1.710 0.464	 1.281 0.730 0.324	
QSC			 1.000 0.000 0.000	 1.329 0.548 0.346	No area distortions. Lowest maximum angular distortion. Expensive. \mathcal{C}^1 discontinuities at face diagonals.
Tangent Adjustment			 1.008 0.199 0.089	 1.234 0.729 0.280	Comparably low distortions. Simple.
Nowell's Method			 1.001 0.205 0.034	 1.235 0.727 0.295	Comparably low distortions. Works on cartesian coordinates.
Continuous Cube			 1.030 0.592 0.183	 1.179 0.728 0.224	Comparably high area distortions. Expensive.
UniCube			 1.001 0.171 0.038	 1.288 0.728 0.348	Low area distortions. High angular distortion RMSE.
Everitt's Method with $c = 1.4511$			 1.003 0.183 0.053	 1.298 0.860 0.349	Low area distortions. High maximum angular distortion.

Table 1 Comparison of cube map variants proposed in the literature.

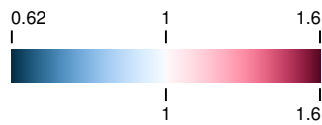


Fig. 2 Color map for D_A (top) and D_I (bottom) in Tabs. 1 and 2. Note that $D_I \geq 1$. Values outside the range displayed here are clamped in Tabs. 1 and 2.

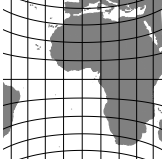
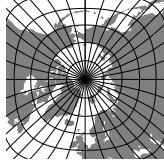


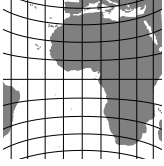
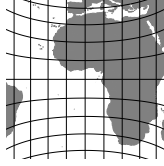


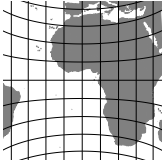
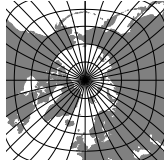

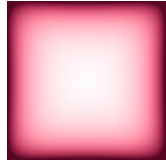
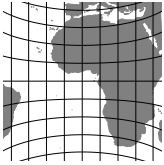
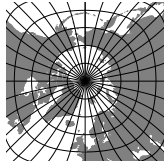

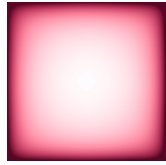
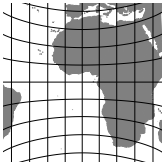
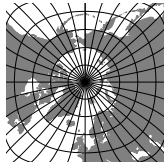
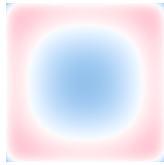
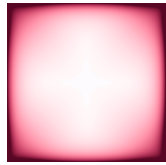
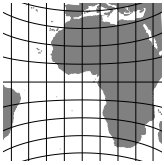
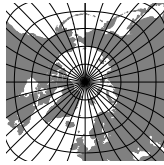

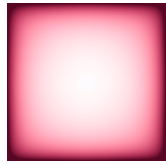
Method	Example cube face at $x=1$	Example cube face at $z=1$	D_A (ideal: 1) avg max err RMSE	D_I (ideal: 1) avg max err RMSE	Remarks
Tangent Sigmoid with $c = 1.1823$ [32]			 1.002 0.143 0.041	 1.290 0.728 0.345	c chosen to minimize area distortion RMSE.
Tangent Sigmoid with $c = 1.1968$			 1.002 0.138 0.041	 1.295 0.728 0.351	c chosen to minimize the maximum area distortion error.
Algebraic Sigmoid with $c = 0.8700$ (recommended method)			 1.002 0.117 0.044	 1.271 0.728 0.327	Lowest maximum area distortion error. Simple and cheap computation.
Logistic Sigmoid with $c = 1.9443$			 1.003 0.119 0.055	 1.257 0.728 0.314	Comparable to algebraic adjustment.
Smoothstep Sigmoid with $c = 0.7507$			 0.992 0.206 0.101	 1.207 0.755 0.280	Worst of the sigmoid function candidates.
Hyperbolic Tangent Sigmoid with $c = 0.9721$			 1.003 0.119 0.055	 1.257 0.728 0.314	Comparable to algebraic adjustment.

Table 2 Comparison of parameterized sigmoid adjustment functions. The top row shows the tangent adjustment variant proposed by Zucker and Higashi [32] (Sec. 4.3). The following rows show sigmoid function variations with parameter c chosen to minimize the maximum area distortion error (Sec. 4.8).

5 Evaluation

To analyse the area and angular distortions of all cube map variants, we use the measures D_A and D_I [15] derived from Tissot's Indicatrix [26]. The basic idea of these measurements is that a map projection maps an infinitesimal circle on the sphere onto an infinitesimal ellipse on the map. From the semi-major axis a and semi-minor axis b of this ellipse, several measurements of projection quality can be derived, e.g. the local scale factor $s = ab$ which is equivalent to the determinant of the Jacobian matrix.

$$\begin{aligned} D_A &= \frac{ab}{R}, & R &= \frac{4}{\frac{4\pi}{6}} = \frac{6}{\pi} \\ D_I &= \frac{a}{b} \end{aligned} \quad (8)$$

R is the ratio between the area of a cube face and the sixth part of the sphere surface area, and is used for normalization. The optimal value for both D_A and D_I is 1. Since $a \geq b$, $D_I \geq 1$.

These two distortion measurements are color-coded as shown in Fig. 2. Note that in Tabs. 1 and 2, values outside the range covered by the color map are clamped.

The results of the distortion evaluations, along with example maps, are listed in Tab. 1 (for all methods previously proposed in the literature) and Tab. 2 (for all tested alternative sigmoid functions). For the sigmoid functions, we manually varied parameter c to minimize the maximum area distortion.

The QSC method is area preserving and has the lowest maximum angular distortion of all tested methods, making it the method with the highest quality by a wide margin. However, its complex and costly computations limit its usefulness for interactive applications.

The sigmoid adjustment functions are simpler and cheaper to compute than the alternative methods proposed in the literature (with the exception of Everitt's method), and they all achieve a lower maximum angular distortion error. They all perform similarly, with the same maximum angular distortion error that is largely independent of the parameter c , and comparable area and angular distortions.

We recommend the algebraic sigmoid adjustment function with parameter $c = 0.87$ since it is simple, cheap to compute, and has the lowest maximum area distortion error while both area and angular distortion RMSE are in an acceptable range.

Furthermore, we tested the numerical accuracy of consecutive forward and inverse transformation for each point in a regular grid of size 512×512 representing one cube face. Since all methods listed in this paper have an analytical inverse, the distance between original point and the result of consecutive transformation

is very small, in the order of 10^{-6} for single precision floating point, except for the Nowell method, where it is in the order of 10^{-4} .

6 Conclusion

This paper gives an overview of alternative cube mapping methods used in interactive Computer Graphics applications. Most of the methods proposed in the literature, e.g. Continuous Cube and Unicube, are outperformed by simpler methods found via systematic evaluation of sigmoid functions, e.g. the algebraic sigmoid function. The best performing mapping method is QSC, which preserves area and additionally limits angular distortions. Since its computations are rather complex and costly, the sigmoid functions provide a good compromise for interactive applications. However, the exact effects of different cube map methods on rendering speed and other system attributes depend on the application area and need to be evaluated in the application context.

Supplementary Material

The supplementary material includes C++ code implementing all mapping methods, along with analysis code that can be used to reproduce all results shown in this paper.

Compliance with Ethical Standards

Conflict of Interest: The author declares that he has no conflict of interest.

Acknowledgements The polygonal world map data used in the example maps in Tabs. 1 and 2 is provided by Bjorn Sandvik, http://thematicmapping.org/downloads/world_borders.php, license CC BY-SA 3.0.

References

1. Arvo, J.: Stratified sampling of 2-manifolds. In: SIGGRAPH Course Notes (2001)
2. Bitterli, B., Novák, J., Jarosz, W.: Portal-masked environment map sampling. *Computer Graphics Forum* **34**(4), 13–19 (2015). DOI 10.1111/cgf.12674
3. Calabretta, M., Greisen, E.: Representations of celestial coordinates in FITS. *Astronomy & Astrophysics* **395**(3), 1077–1122 (2002). DOI 10.1051/0004-6361:20021327
4. Calabretta, M.R., Roukema, B.F.: Mapping on the HEALPix grid. *Monthly Notices of the Royal Astronomical Society* **381**(2), 865–872 (2007). DOI 10.1111/j.1365-2966.2007.12297.x

5. 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 (1975). URL <https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/ADA010232.xhtml>
6. Dimitrijević, A., Lambers, M., Rančić, D.: Comparison of spherical cube map projections used in planet-sized terrain rendering. *Facta Universitatis, Series: Mathematics and Informatics* **31**(2), 259–297 (2016). URL <http://casopisi.junis.ni.ac.rs/index.php/FUMathInf/article/view/871>
7. Everitt, C.: "projection" repository (2016). URL <https://github.com/casseveritt/projection/>
8. Gascuel, J.D., Holzschuch, N., Fournier, G., Péroche, B.: Fast non-linear projections using graphics hardware. In: *Proc. Symp. Interactive 3D Graphics and Games (I3D)*, pp. 107–114 (2008). DOI 10.1145/1342250.1342267
9. Greene, N.: Environment mapping and other applications of world projections. *IEEE Comp. Graph. and Applications* **6**(11), 21–29 (1986). DOI 10.1109/MCG.1986.276658
10. Grimm, C.M., Niebruegge, B.: Continuous cube mapping. *Journal of Graphics, GPU, and Game Tools* **12**(4), 25–34 (2007). DOI 10.1080/2151237X.2007.10129250
11. Harrison, E., Mahdavi-Amiri, A., Samavati, F.: Optimization of inverse snyder polyhedral projection. In: *Int. Conf. on Cyberworlds*, pp. 136–143 (2011). DOI 10.1109/CW.2011.36
12. Ho, T.Y., Wan, L., Leung, C.S., Lam, P.M., Wong, T.T.: Unicube for dynamic environment mapping. *IEEE Trans. Visualization and Computer Graphics* **17**(1), 51–63 (2011). DOI 10.1109/TVCG.2009.205
13. Kemen, B., Hrabcak, L.: Outerra (2014). URL <http://www.outerra.com>
14. 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), 719–733 (2009). DOI 10.1109/TVCG.2009.43
15. Lambers, M.: Mappings between sphere, disc, and square. *Journal of Computer Graphics Techniques* **5**(2), 1–21 (2016). URL <http://jcgt.org/published/0005/02/01/>
16. Lambers, M., Kolb, A.: Ellipsoidal cube maps for accurate rendering of planetary-scale terrain data. In: *Proc. Pacific Graphics (Short Papers)*, pp. 5–10 (2012). DOI 10.2312/PE/PG/PG2012short/005-010
17. Lambers, M., Sommerhoff, H., Kolb, A.: Realistic lens distortion rendering. In: *Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)* (2018). URL <http://wscg.zcu.cz/wscg2018/2018-WSCG-Papers-Separated.html>
18. Lee, L.: Conformal projections based on jacobian elliptic functions. *Cartographica: The Int. J. for Geographic Information and Geovisualization* **13**(1), 67–101 (1976). DOI 10.3138/X687-1574-4325-WM62
19. Lerbour, R., Marvie, J.E., Gautron, P.: Adaptive real-time rendering of planetary terrains. In: *Proc. Int. Conf. Computer Graphics, Visualization and Computer Vision (WSCG)* (2010). URL http://wscg.zcu.cz/WSCG2010/Papers_2010/!_2010_FULL-proceedings.pdf
20. Nowell, P.: Mapping a cube to a sphere. <http://mathproofs.blogspot.de/2005/07/mapping-cube-to-sphere.html> (2005)
21. 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 (1976). URL <http://www.dtic.mil/docs/citations/ADA026294>
22. Rančić, M., Purser, R.J., Mesinger, F.: A global shallow-water model using an expanded spherical cube: Gnomonic versus conformal coordinates. *Quarterly Journal of the Royal Meteorological Society* **122**(532), 959–982 (1996). DOI 10.1002/qj.49712253209
23. Ronchi, C., Iacono, R., Paolucci, P.S.: The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *J. Computational Physics* **124**(1), 93–114 (1996). DOI <https://doi.org/10.1006/jcph.1996.0047>
24. Roşca, D., Plonka, G.: Uniform spherical grids via equal area projection from the cube to the sphere. *Journal of Computational and Applied Mathematics* **236**(6), 1033–1041 (2011). DOI 10.1016/j.cam.2011.07.009
25. Scherzer, D., Wimmer, M., Purgathofer, W.: A survey of real-time hard shadow mapping methods. *Computer Graphics Forum* **30**(1), 169–186 (2011). DOI 10.1111/j.1467-8659.2010.01841.x
26. Snyder, J.: Map projections—a working manual, *Professional Paper*, vol. 1395. US Geological Survey (1987). DOI 10.3133/pp1395
27. Snyder, J.: An equal-area map projection for polyhedral globes. *Cartographica* **29**(1), 10–21 (1992). DOI 10.3138/27H7-8K88-4882-1752
28. Snyder, J., Mitchell, D.: Sampling-efficient mapping of spherical images (2001). URL <https://www.microsoft.com/en-us/research/publication/sampling-efficient-mapping-spherical-images/>. Microsoft Research Technical Report
29. Various: Mapping a sphere to a cube. <http://stackoverflow.com/questions/2656899/mapping-a-sphere-to-a-cube> (2010)
30. Wan, L., Wong, T.T., Leung, C.S.: Isocube: Exploiting the cubemap hardware. *IEEE Trans. Visualization and Computer Graphics* **13**(4), 720–731 (2007). DOI 10.1109/TVCG.2007.1020
31. Wong, T.T., Wan, L., Leung, C.S., Lam, P.M.: Shader X4: Advanced Rendering Techniques, chap. Real-Time Environment Mapping with Equal Solid-Angle Spherical Quad-Map, pp. 221–233. Charles River Media (2006)
32. Zucker, M., Higashi, Y.: Cube-to-sphere projections for procedural texturing and beyond. *Journal of Computer Graphics Techniques (JCGT)* **7**(2), 1–22 (2018). URL <http://jcgt.org/published/0007/02/01/>