

COMPARISON OF SPHERICAL CUBE MAP PROJECTIONS USED IN PLANET-SIZED TERRAIN RENDERING

Aleksandar M. Dimitrijević, Martin Lambers and Dejan D. Rančić

Abstract. A wide variety of projections from a planet surface to a two-dimensional map are known, and the correct choice of a particular projection for a given application area depends on many factors. In the computer graphics domain, in particular in the field of planet rendering systems, the importance of that choice has been neglected so far and inadequate criteria have been used to select a projection. In this paper, we derive evaluation criteria, based on texture distortion, suitable for this application domain, and apply them to a comprehensive list of spherical cube map projections to demonstrate their properties.

Keywords: Map projection, spherical cube, distortion, texturing, graphics

1. Introduction

Map projections have been used for centuries to represent the curved surface of the Earth with a two-dimensional map. A wide variety of map projections have been proposed, each with different properties. Of particular interest are scale variations and angular distortions introduced by map projections – since the spheroidal surface is not developable, a projection onto a plane cannot be both conformal (angle-preserving) and equal-area (constant-scale) at the same time. These two properties are usually analyzed using Tissot’s indicatrix. An overview of map projections and an introduction to Tissot’s indicatrix are given by Snyder [24].

In computer graphics, a map projection is a central part of systems that render planets or similar celestial bodies: the surface properties (photos, digital elevation models, radar imagery, thermal measurements, etc.) are stored in a map hierarchy in different resolutions. During the rendering, the data from this map hierarchy are sampled for display on a screen. Despite its central role, many systems do not pay much attention to the choice of projection for the map hierarchy. Often, a relatively straightforward approach is used, which leads to sampling problems both during the creation of the map hierarchy and during its sampling at rendering time.

Received February 19, 2015; accepted January 18, 2016

2010 *Mathematics Subject Classification.* Primary 68U05; Secondary 65D18, 68W25

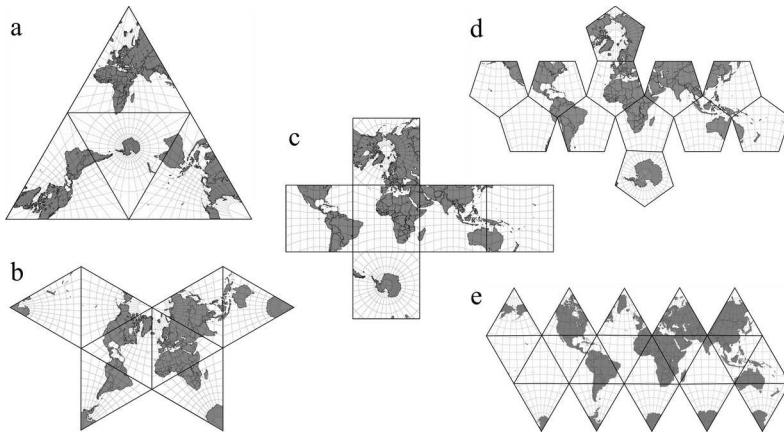


FIG. 1.1: Polyhedral projections based on Platonic solids: tetrahedron (a), octahedron (b), hexahedron (c), dodecahedron (d) and icosahedron (e).

This is particularly apparent with systems that use a single map to cover the whole planet surface. Such systems usually exhibit strong distortions and artifacts in the polar regions. Examples include the well-known commercial products like Google Earth [5] and NASA World Wind [15]. A few systems do care about the projection they use, but use insufficient evaluation criteria and/or evaluate too few alternatives to make a good choice [10, 13, 11].

Projecting a spheroidal surface to a single plane (flat or folded into a cylinder or a cone) always results in singularities [10, 24], therefore the first step for improvement is to subdivide the spheroidal surface into several regions, each of which is projected to a separate projection plane. The subdivision reduces map distortion, but increases the number of interrupts. The faces of an encompassing or inscribed polyhedron are very good candidates for the projection planes, hence the polyhedral projections have been used for centuries to represent the surface of the Earth. Fig. 1.1 displays unfolded polyhedral projections based on Platonic solids.

As it can be seen in Fig. 1.1, the increase of the polyhedral faces number reduces distortion and increases interrupts at the same time. The number of interrupts is also an important aspect of a map projection. For paper maps, interrupts make visual discontinuities, while for electronic maps (i.e. textures) they may require separate data sets for each region. The number of data sets may have a direct impact on the memory usage [4]. Therefore, it should be minimized if possible.

For the purposes of computer graphics, the projection to the faces of a cube (as a special form of a hexahedron) is of particular interest because each face is rectangular and thus allows straightforward storage of map data in common 2D file formats, as well as management of rendering data in common 2D texture formats. Also, cube based projections expose moderate distortion and number of interrupts.

In this paper, we derive a set of evaluation criteria, based on texture distortion,

and apply them to a comprehensive list of hexahedral map projections to demonstrate their properties. This list covers all spherical cube map projection known to be used in planet-sized terrain rendering. One of the projections (Outerra Spherical Cube Map) is published for the first time, thanks to its original implementer Brano Kemen [8]. We implemented all of these projections as well as map projection software and a set of evaluation tests.

The remainder of this paper is organized as follows. Sec. 2. gives an overview of related work in the field of planet rendering, with an emphasis on the choice of map projections. In Sec. 3., we derive the evaluation criteria that we apply to spherical cube map projections, reviewed in Sec. 4. The results of this evaluation are presented and discussed in Sec. 5. Finally, Sec. 6. concludes the paper.

2. Related Work

Map projections from a sphere surface to a plane have a long history, and a wide variety of methods have been developed, each with specific properties carefully chosen for specific tasks. An overview is given by Snyder [24].

A popular map projection for planet rendering systems, including the commercial offerings Google Earth [5] and NASA World Wind [15], is the equidistant cylindrical (or plate carrée) projection. Like all single-map projections, it suffers from singularities. In proximity to the poles, very small surface areas are mapped to many samples on the map, distributed over elongated areas. This causes significant storage and data access overhead in the renderer as well as a radial blur in the rendered image [10].

The problems associated with singularities can only be avoided by subdividing the sphere and using multiple maps. Kooima et al. use equidistant cylindrical projection for the equatorial part of the planet and two additional polar stereographic projections for the polar regions. Weighted averages are used for smooth transitions between the three regions [10].

Among polyhedral projections, the cube based approaches are very popular. They divide the spherical surface into six identical regions, as shown in Fig. 2.1. This allows using a single map projection (Fig. 2.2) that behaves consistently at cube face borders, thus eliminating the need for weighted averaging. Furthermore, the rectangular maps for the cube faces allow straightforward data storage using quadtree hierarchies and common file formats and straightforward data management in the rendering system using common and efficient rectangular textures.

The straightforward projection of the sphere to the cube faces is a gnomonic projection. The distortions introduced by gnomonic projection onto cube faces are significant. For this reason, Lerbour and et al. proposed an adjustment to the gnomonic projection [13] that reduces these distortions to some degree. Lammers and Kolb compared the gnomonic and adjusted gnomonic projections with the Quadrilateralized Spherical Cube (QSC) projection, and chose the latter [11]. As we will demonstrate in the next sections, their evaluation was too limited.

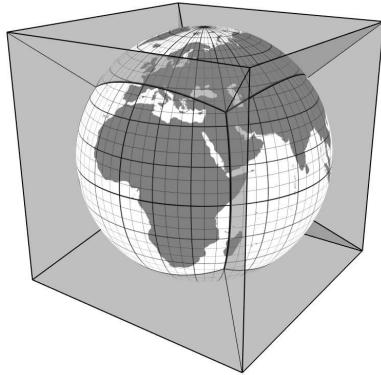


FIG. 2.1: A spherical planet model inscribed into a cube. The cube partitions the sphere surface into six equal areas.

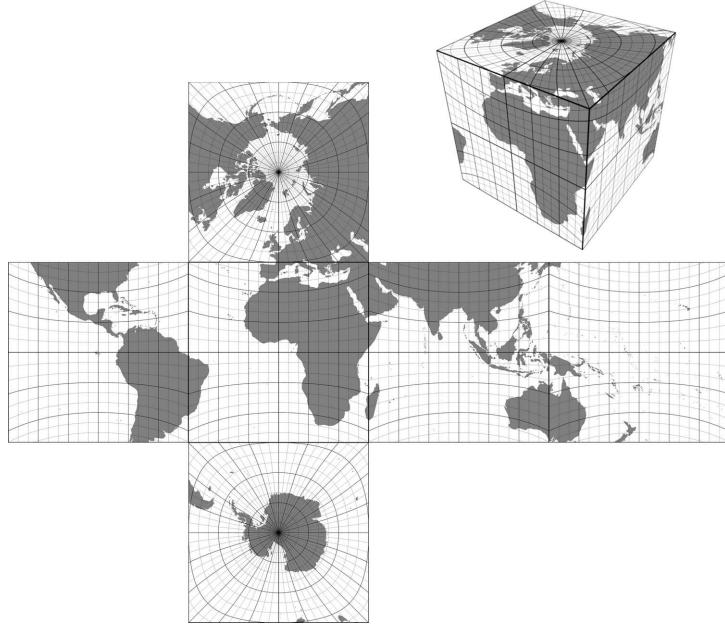


FIG. 2.2: The world mapped using a cube map projection.

In the application area of planet rendering, very few map projections have been considered for subdivisions of the planet. Of the cube-based subdivisions, only the gnomonic projection, an adjusted version of the gnomonic projection and the QSC projection are documented in the literature. In the next sections, we extend this list with a projection used in the Outerra rendering engine [8], an approximately equal-area projection based on the sphere representation in the Cartesian coordinates

[17] and a variant of the HEALPix projection [6]. All of these spherical cube map projections are explained in Sec. 4. and compared using the criteria derived in Sec. 3., according to the distortion they introduce in the texture application for planet-sized terrain rendering algorithms.

3. Evaluation Approach

In classical applications of map projections, the two-dimensional map is the final product and intended for direct use by the end user. In a planet rendering application, on the other hand, the map is just an intermediate data representation. Consequently, the projection is used in two steps: first, when mapping the original data to the cube-based hierarchical representation in a preprocessing step, and second, when sampling this representation during rendering to produce the end result. While a poor choice of map projection can have negative effects already during preprocessing, the crucial step for the quality of the end result is the rendering step.

We will, therefore, focus on rendering and sampling aspects of spherical cube map projections. To this end, we first examine, in Sec. 3.1., the way textures are applied to a rendered terrain. This discussion provides sufficient details to understand how texture filtering is performed and how effects of distortion can be reduced. In Sec. 3.2., we discuss how applied projections introduce a texture distortion, while Sec. 3.3. explains the methods used in the evaluation process. The main evaluation criterion for the projection comparison is texture distortion, but we also consider the precision and efficiency of the forward and inverse transformations, as well as the size of the applied textures.

3.1. Texture Application

Two-dimensional textures are image-overlays applied to geometrical objects to improve their fidelity without increasing their complexity. A texture application entails mapping from texture space to screen space. The mapping is done through two filtering schemes: *minification* and *magnification* [23]. When a texel (the smallest unit of a texture) is smaller than an area, it is applied to (one texel maps to multiple pixels on a screen) a magnification filter is used. Otherwise, multiple texels are mapped to a single pixel using a minification filter. In order to minimize the aliasing effect caused by minification, multiple levels of detail (the same texture in different resolutions) are used. That enables choosing the level where the texel-to-pixel ratio is near to one. Lance Williams proposed mipmaps as an efficient way to pack multiple levels of detail into a single texture [28]: each lower resolution level is constructed from the higher resolution level by downsampling with a factor of two in both horizontal and vertical directions.

Mipmaps have been used for decades as a very efficient way of texture mapping. As the size of the object being mapped increases, however, the application of high-fidelity mipmapped textures becomes untenable. The visualization of the planet

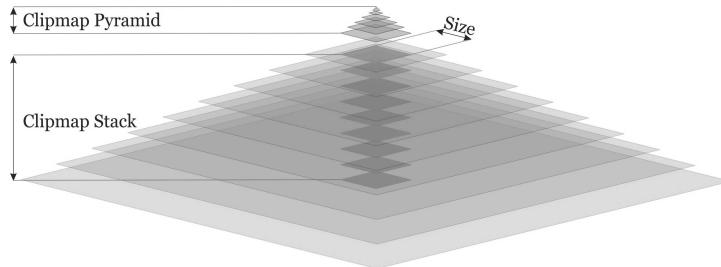


FIG. 3.1: A clipmap – an updatable partial mipmap. Clipmap levels are grouped into two sets: the clipmap pyramid (low-resolution static levels) and the clipmap stack (higher-resolution dynamic levels). The levels in the clipmap stack are centered on the focal point and toroidally updated as it moves.

Earth, with submeter accuracy, could require several petabytes of storage space for the full mipmapped texture; clearly this is too much to fit in the graphics or system memory. To solve this problem, several techniques have been developed. One of the most popular is known as *clippmapping* [26]. A clipmap is an updatable representation of a partial mipmap, in which each mipmap level is clipped to a specified size. Instead of the exponential growth of full mipmaps, a clipmap grows linearly with each new level of detail.

The appropriate level of clipmap to apply is chosen according to the texture scale-factors. The scale-factors of the applied texture are calculated using partial derivatives of the given functions $u(x, y)$ and $v(x, y)$ that map screen coordinates (x, y) to the two-dimensional texture coordinates s and t , respectively. Depending on the orientation of the surface, scale-factors along the horizontal and vertical screen axes (ρ_x and ρ_y , respectively) may differ.

$$(3.1) \quad \begin{aligned} \rho_x(x, y) &= \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2} \\ \rho_y(x, y) &= \sqrt{\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2} \\ \rho_{max}(x, y) &= \max(\rho_x(x, y), \rho_y(x, y)) \\ \rho_{min}(x, y) &= \min(\rho_x(x, y), \rho_y(x, y)) \end{aligned}$$

Since derivatives may be computationally expensive and/or numerically unstable [9], they are usually approximated in the graphics hardware by computing forward/backward differences between neighboring pixels in a 2×2 block. Whether the forward or backward difference is used depends on the position of the pixel in the block. In standard texture filtering schemes, a proper level of detail (λ) is selected according to $\rho_{max}(x, y)$ which gives smoother results:

$$(3.2) \quad \lambda(x, y) = \log_2(\rho_{max}(x, y))$$

Further smoothing is achieved by linear interpolation of the texel values closest to the center of the displayed pixel. Bilinear filtering interpolates values of the texel from a single level only, selected by the rounded value of an integral part of λ . Trilinear filtering further improves the smoothness by combining two adjacent levels. The integral part of λ selects a more detailed level, while the fractional part defines blending factor with the next coarser one ($\lambda + 1$).

If the surface being viewed is at an oblique angle, trilinear filtering could result in a blurry display. The fidelity and sharpness of the applied texture, in that case, can be improved through anisotropic filtering [19]. Unlike the previous (isotropic) filtering schemes, where the footprint of the filter projection into texture space is a square, anisotropic filtering may have very narrow or long footprints. A higher degree of anisotropy may improve texture filtering quality, but at the same time reduce the texture filtering rate. Hence, the maximum degree of anisotropy (ϱ) is always limited, usually to 16. According to the OpenGL anisotropic texture filter specification [19], a proper texture level for anisotropic filtering should be selected using the following equations:

$$(3.3) \quad \begin{aligned} N(x, y) &= \min \left(\left\lceil \frac{\rho_{\max}(x, y)}{\rho_{\min}(x, y)} \right\rceil, \varrho \right) \\ \lambda(x, y) &= \log_2 \left(\frac{\rho_{\max}(x, y)}{N(x, y)} \right) \end{aligned}$$

In the next section, we shall see how applied filtering is used to minimize the manifestation of texture distortion.

3.2. Distortion

Tissot's indicatrices are very useful in estimating the distortion of a projection of the Earth's surface to a planar map. They are used in cartography to evaluate the size and shape of the objects depicted on the map. However, in computer graphics the more important consideration is the distortion of the textures after application to a 3D model of the planet. In order to evaluate this texture application distortion we will introduce two measures of distortion:

- the texel aspect distortion and
- the texel area distortion.

The texel aspect distortion (δ_{aspect}) is defined as the texel width (Λ_x) to height (Λ_y) ratio after unprojecting to the surface of a planet:

$$(3.4) \quad \delta_{\text{aspect}} = \frac{\Lambda_x}{\Lambda_y}$$

In Eq. 3.4, Λ_x and Λ_y are calculated as distances on the spheroidal surface along directions aligned with the texture (i.e. projection) axes X and Y, respectively. The calculation is based on the central differences using the following equations:

$$(3.5) \quad \begin{aligned} (\varphi_l, \theta_l) &\leftarrow \text{inverse}(x - \Delta/2, y) \\ (\varphi_r, \theta_r) &\leftarrow \text{inverse}(x + \Delta/2, y) \\ (\varphi_b, \theta_b) &\leftarrow \text{inverse}(x, y - \Delta/2) \\ (\varphi_t, \theta_t) &\leftarrow \text{inverse}(x, y + \Delta/2) \\ \Lambda_x &= \sigma(\varphi_l, \theta_l, \varphi_r, \theta_r) \\ \Lambda_y &= \sigma(\varphi_b, \theta_b, \varphi_t, \theta_t) \end{aligned}$$

In the previous equations, x and y are the coordinates of the point in texture space and Δ is a texel size. The function *inverse* depends on the chosen projection. Sec. 4. presents all spherical cube map projections with their forward transformation (from a sphere onto a plane, with normalized coordinates in the range [-1,1]) and inverse transformation (from the plane back to the sphere). A distance on the sphere between two points defined by their spherical coordinates (φ_1, θ_1) and (φ_2, θ_2) , is calculated with the function σ based on the following formulae:

$$(3.6) \quad \begin{aligned} \sigma(\varphi_1, \theta_1, \varphi_2, \theta_2) &= \\ 2R_e \arcsin \left(\sqrt{\sin^2 \frac{|\theta_1 - \theta_2|}{2} + \cos \theta_1 \cos \theta_2 \sin^2 \frac{|\varphi_1 - \varphi_2|}{2}} \right) \\ R_e &= \frac{2 \cdot a + b}{3} \end{aligned}$$

Since the Earth is an oblate spheroid with very small flattening ($f = 1/298.257223563$), in order to simplify equations throughout this paper, we are using a spherical approximation with the same volume as the reference ellipsoid ($R_e = (a \cdot b)^{1/3} \approx (2 \cdot a + b)/3 = 6371\text{km}$). The parameters a and b in the previous equations refer to the semi-major and semi-minor axes of the WGS84 ellipsoid [16], respectively. For the ellipsoidal model of the planet, the distance between two points can be calculated more accurately using a Vincenty's inverse method [27]. However, the difference between the great circle distance formula (Eq. 3.6) and Vincenty's inverse formula in calculating distortion values is negligible. For example, in the case of QSC projection (Sec. 4.5.), the relative error of the maximum texel aspect distortion of the spherical approximation (compared to the WGS84 ellipsoidal model) is only about $5 \cdot 10^{-6}$. Therefore, the use of spherical approximations in the following discussion is justified.

Although the texel aspect distortion is usually neglected when a projection is chosen, the impact on the rendered surface can be significant and it manifests through:

- aliasing/blurring,
- additional anisotropy and
- a requirement for bigger textures.

The aliasing or blurring effects are consequences of shrinking or stretching of the texture over the applied surface. A texel distortion is always the combination of both aspect and area distortion. In order to simplify the analysis, without loss of generality, let us assume that distortion affects just a single direction (Fig. 3.2). If $\delta_{\text{aspect}} > 1$, stretching appears, while $\delta_{\text{aspect}} < 1$ causes texture shrinking. Depending on the texture scale-factors (Eq. 3.1), if the distortion is significant, even the current level of a clipmap (λ) may change. Without bilinear or trilinear filtering, the distortion results in a noticeable aliasing effect. Aliasing effects can be reduced by bilinear/trilinear filtering, but only to a certain degree and with an accompanying blur effect (Fig. 3.2).

Since aspect distortion introduces uneven texture sampling along different axes, a higher texture fidelity and sharpness can be achieved only by using anisotropic filtering (Fig. 3.2), but this introduces computational costs and is limited by the maximum anisotropy degree. As the surface is viewed from a more oblique angle, a higher degree of anisotropic filtering is required to provide sharpness. Because part of the available anisotropy is spent on aspect distortion correction, oblique surfaces may appear blurry.

Another reason for minimizing texel aspect distortion, even more important than the blur of oblique surfaces, is the need for bigger textures. Because of aspect distortion, after the application to a terrain, a texture changes its aspect and cover the different area along different directions, resulting in more details in the direction where shrinking occurs and fewer details in the direction where stretching occurs. If we select the texture level with enough details for the stretching direction while not taking into account aspect distortion, it will result in exceeding the size of a current clipmap level along the direction where shrinking occurs (black strips in Fig. 3.2). This issue can be solved by using texture levels bigger than their nominal size, for the factor greater or equal to the maximum texel distortion. Bigger textures induce higher memory consumption and longer update times. Choosing a coarser clipmap level, to avoid exceeding the size of the current level, leads to blurry rendering results.

The texel area distortion (δ_{area}) is defined as the ratio of the texel size at the current position ($\Lambda_x \cdot \Lambda_y$) and the texel size at the center of the cube face ($\Lambda_0 \cdot \Lambda_0$) after unprojecting to the surface of a planet:

$$(3.7) \quad \delta_{\text{area}} = \frac{\Lambda_x \Lambda_y}{\Lambda_0 \Lambda_0}$$

Unlike texel aspect distortion, the texel area distortion is important only if it changes across the surface of the current clipmap level λ . If δ_{area} is nearly constant, a distortion actually does not exist. The texel area distortion in non-equal-area

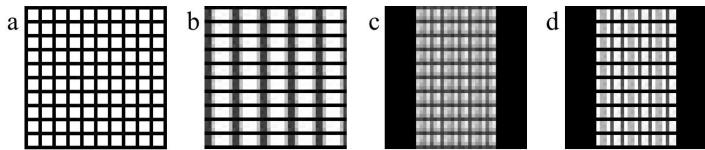


FIG. 3.2: Effects of the texel aspect distortion. From left to right: texture without distortion (a), horizontal stretching $\delta_{\text{aspect}} = 2$ (b), horizontal shrinking with trilinear filtering $\delta_{\text{aspect}} = 0.67$ (c), horizontal shrinking with anisotropic filtering $\delta_{\text{aspect}} = 0.67$ (d)

projection can be nearly constant only for the higher resolution levels. Such levels have smaller spatial extent, which prevents significant change in the value of area distortion. For the lower resolution levels, where it is not the case, area distortion has to be treated the same way as aspect distortion.

At the first glance, it seems that the texel area distortion does not require bigger textures, but only a modification of the clipmap level selection, defined by the following equation:

$$(3.8) \quad \lambda' = \lambda + \log_2 (\sqrt{\delta_{\text{area}}})$$

However, since texel area distortion gradually changes across the surface and its value is usually not high enough to switch to the coarser clipmap level, texel area distortion also contributes to the need for larger textures. If the value of λ' is clamped to a higher integral value, a nominal size of the texture can be used, but the blurry outcome is inevitable. In order to preserve the ability to properly blend adjacent clipmap levels and gain required sharpness of the visualization, we have to provide levels big enough to contain nominal spatial extent, no matter where the viewer is located.

Considering both texel aspect and area distortion, the size of the storage space for a clipmap level should be bigger than its nominal size for the factor ε , where:

$$(3.9) \quad \varepsilon = \max(\delta_{\text{aspect}}^{\max}, \delta_{\text{area}}^{\max})$$

$\delta_{\text{aspect}}^{\max}$ and $\delta_{\text{area}}^{\max}$ are the maximum values of the texel aspect and area distortion, respectively.

In the conclusion, both texel aspect and area distortions result in higher texture storage demands and, hence, slightly slower update. The texel aspect distortion also spends a part of the available anisotropic filtering range. The higher aspect distortion the blurrier the display of a surface viewed from an oblique angle. The texel distortion elaborated in this section is one of the very important aspects that has to be evaluated when a proper projection for the terrain rendering system is chosen. The following section presents an evaluation method we used for comparing spherical cube map projections in this paper.

3.3. Evaluation Method

In order to examine their properties, we have implemented all spherical cube map projections that we could collect. Our development environment was the Microsoft Visual Studio 2013 and we used C/C++ compiler. All calculations are done on the CPU using double extended precision [7]. For each face of the spherical cube, the following tests were carried out:

- precision self-test,
- forward/inverse transformation time measurement,
- texel aspect distortion statistics, and
- texel area distortion statistics.

The precision self-test was used both for an internal implementation check and for checking the quality of the projection. For each of N^2 equally spaced points in the map space ($x, y \in [-1, 1]$), the inverse transformation is executed followed by forward transformations of its results. The final results should match the starting coordinates. The difference of the starting and final values projected to the surface of the Earth is used as a precision evaluation criterion.

Projection (forward transformation) and unprojection (inverse transformation) are required in data preparation and during terrain rendering, respectively, which is why the transformation execution time is an important property for evaluation. High-performance counters [14] are used for measuring the transformation of huge matrices with input coordinates. The measured time, besides the transformations themselves, includes iteration and matrix access time. Such overhead is required in order to prevent compiler optimizations from leading to incorrect results.

The texel aspect and area distortion are computed, using Eq. 3.4 through 3.7, for the points on the equally spaced grid in map space. The statistics discussed in this paper are computed for the set of up to 8000^2 points, with $\Delta = 1 \cdot 10^{-6}$.

In addition to the numerical tests, the same software was used to generate world maps in each of the projections, to reproject data from the equidistant cylindrical projection and to produce the images of texel aspect and area distribution over the faces of the cube. The results of the tests, as well as the generated images, are displayed and discussed in the following sections.

4. Spherical Cube Map Projections

This section presents a comprehensive set of Spherical Cube Map (SCM) projections. All projections are presented with their forward and inverse transformations, visual aspects of distortion shown through continents coastline and graticule skewing for the *front* and *top* faces, and the distribution of texel aspect and area distortions over the faces of the cube. The given formulae for forward and inverse

transformations apply to the cube face centered on $\varphi = 0$ and $\theta = 0$ (the *front* face). An exception is the HEALPix projection, where transformations differ for equatorial and polar regions, and, hence, have to be treated separately. Also, for each of the projections, the effects of the inverse transformation are visualized by reprojecting continent coastline and equally-spaced regular grid in the map space back to the globe.

4.1. Tangential Spherical Cube

Tangential Spherical Cube (TSC) is the simplest SCM projection. It uses the standard gnomonic projection to map the globe onto the six faces of a tangent cube. As a gnomonic projection, it is distortion-free only at the point where the tangent plane touches the surface. The distortion of shape, area and scale increases with the distance from that point. The gnomonic projection dates from Ancient Greece. Thales of Miletus (624-546 BC) used it for celestial maps. TSC was used for terrestrial maps in the beginning of the 19th century for the first time [24]. It is neither conformal nor equal-area projection.

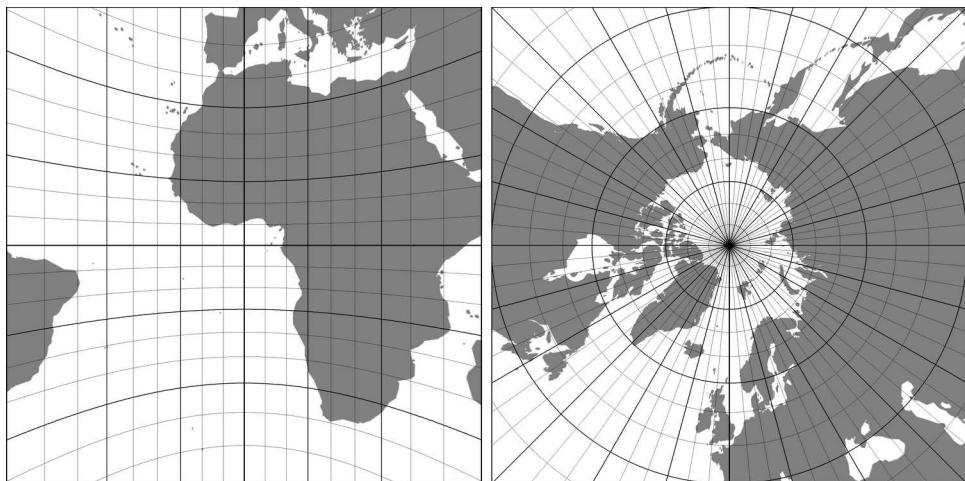


FIG. 4.1: Front and top faces of the TSC projection.

Forward transformation:

$$(4.1) \quad \begin{aligned} x &= \tan \varphi \\ y &= \frac{\tan \theta}{\cos \varphi} \end{aligned}$$

Inverse transformation:

$$(4.2) \quad \begin{aligned} \varphi &= \arctan x \\ \theta &= \arctan(y \cdot \cos \varphi) \end{aligned}$$

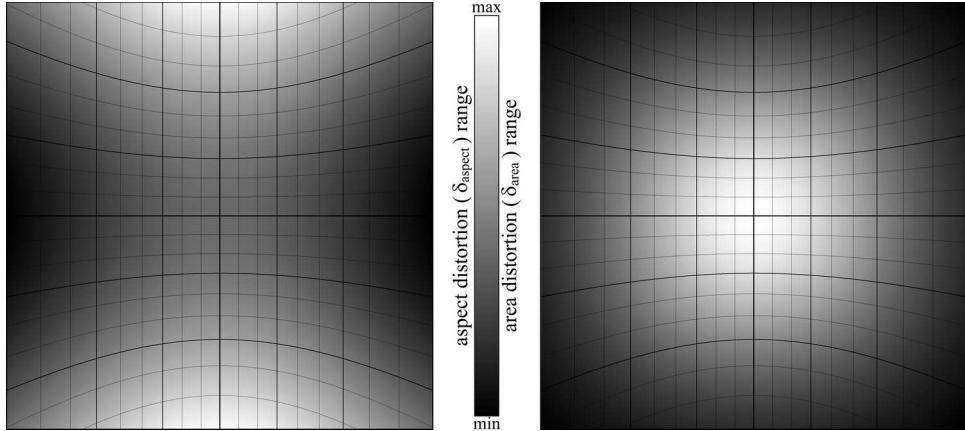


FIG. 4.2: The distribution of the texel aspect (left side) and area (right side) distortion over the face of the cube when TSC projection is used. The texel aspect distortion ranges from 0.707 to 1.414 ($\delta_{\text{aspect}}^{\text{max}}/\delta_{\text{aspect}}^{\text{min}} = 2.0$), while the texel area distortion ranges from 0.222 to 1.0 ($\delta_{\text{area}}^{\text{max}}/\delta_{\text{area}}^{\text{min}} = 4.5$).

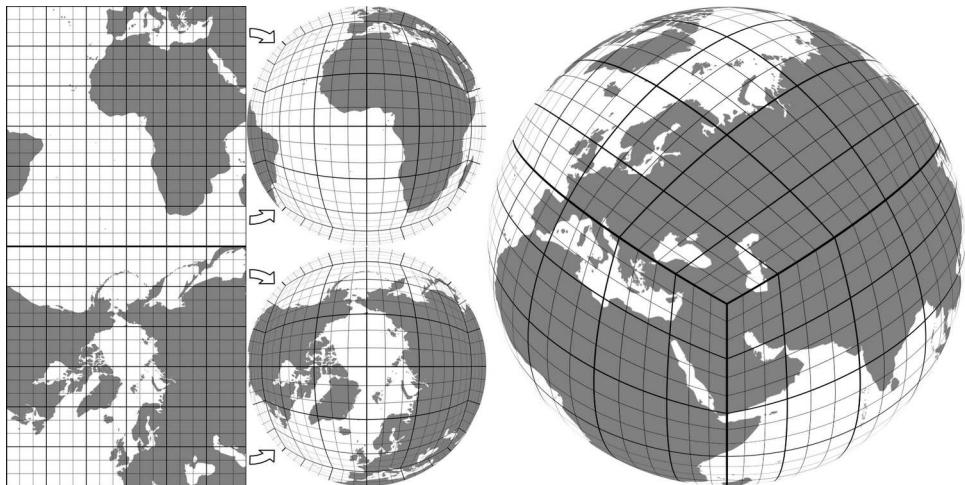


FIG. 4.3: The inverse transformation of the rectangular equidistant grid in TSC planar space and the continent coastlines to a globe surface.

Although it is simple, TSC is rarely used for visualization of the planet Earth, because of its significant distortions, both in aspect and area. Fig. 4.1 depicts a distortion through continent coastlines and graticule skewing, while Fig. 4.2 gives a spatial distribution of distortions over the faces of the cube through the shades of gray. Darker tones for the δ_{aspect} represent shrinking in the X-direction, while brighter tones represent shrinking in the Y-direction. Darker tones for δ_{area} repre-

sent an area shrinking. Texels at the corners of a face of the cube cover 4.5 times smaller surface than texels at the center. The effect of area shrinking can be explicitly visualized through the inverse transformation of the rectangular equidistant grid in the projection space to a globe surface (Fig. 4.3). While aspect distortion is slightly better than the equal-area or approximately equal-area SCM projections, area distortion is far worse.

4.2. Adjusted Spherical Cube

Adjusted Spherical Cube (ASC) modifies TSC in order to reduce area distortion. Instead of sampling the plane of projection, ASC samples the map directly in spherical coordinates with steps expressed in terms of angles [12, 13]. Thus, the forward transformation of TSC can be turned into ASC by simply calculating \arctan of the x and y coordinates and normalizing to ± 1 .

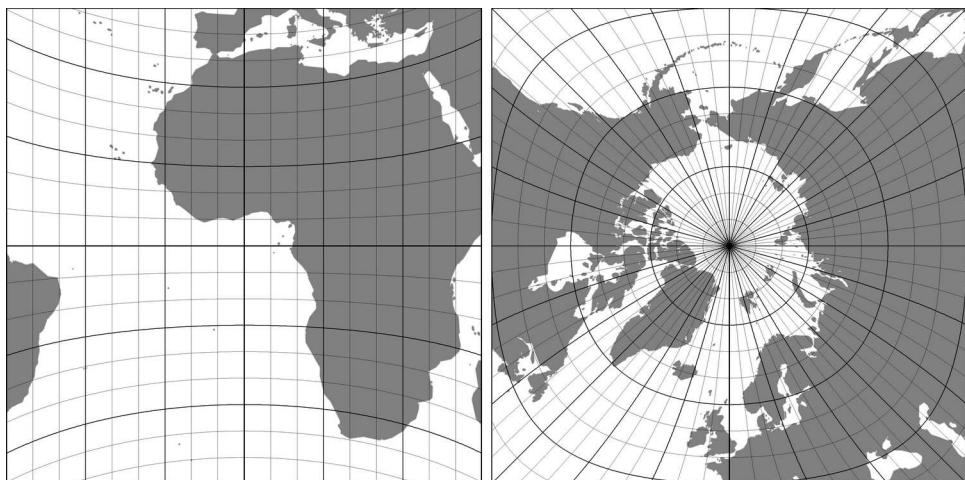


FIG. 4.4: Front and top faces of the ASC projection.

Forward transformation:

$$(4.3) \quad \begin{aligned} x &= \varphi \cdot \frac{4}{\pi} \\ y &= \arctan \left(\frac{\tan \theta}{\cos \varphi} \right) \cdot \frac{4}{\pi} \end{aligned}$$

Inverse transformation:

$$(4.4) \quad \begin{aligned} \varphi &= x \cdot \frac{\pi}{4} \\ \theta &= \arctan \left(\tan \left(\frac{\pi \cdot y}{4} \right) \cdot \cos \varphi \right) \end{aligned}$$

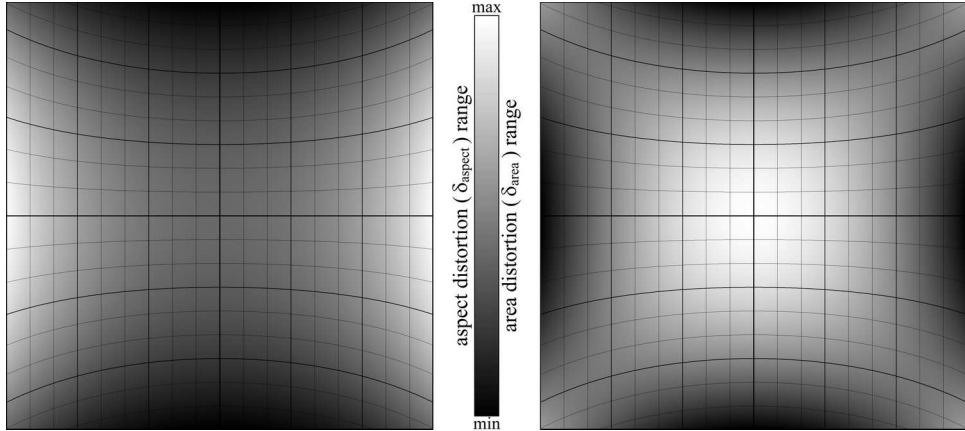


FIG. 4.5: The distribution of the aspect (left side) and area (right side) distortion over the face of the cube when ASC projection is used. The texel aspect distortion ranges from 0.707 to 1.414 ($\delta_{\text{aspect}}^{\max}/\delta_{\text{aspect}}^{\min} = 2.0$), while the texel area distortion ranges from 0.707 to 1.0 ($\delta_{\text{area}}^{\max}/\delta_{\text{area}}^{\min} = 1.414$).

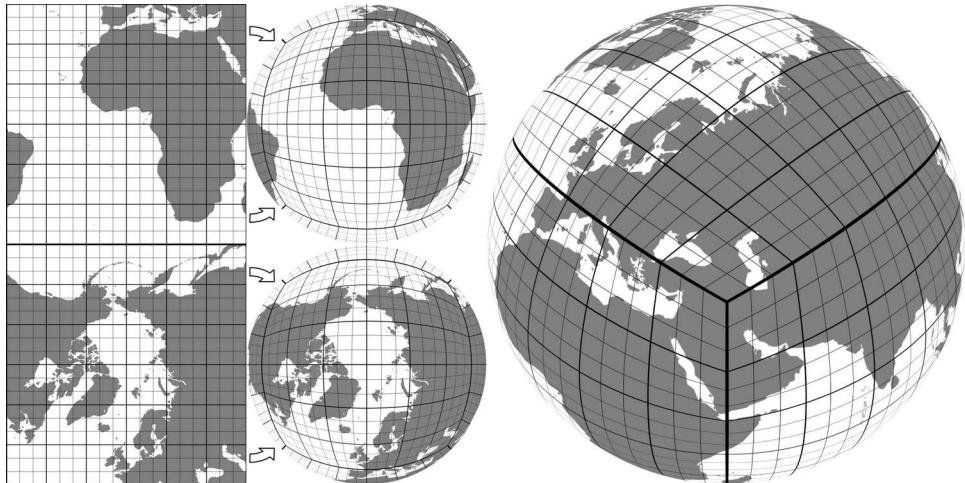


FIG. 4.6: The inverse transformation of the rectangular equidistant grid in ASC planar space and the continent coastlines to a globe surface.

ASC was published for the first time in 1996, with slightly different formulae based on colatitude [22]. It was reinvented and used for the planet-sized terrain rendering many years later, in 2009 [12]. Like TSC, ASC is neither conformal nor equal-area. However, the proposed adjustment of TSC effectively reduces area distortion. Graticule spacing increases toward the midpoints of the edges of the cube (Fig. 4.4), which indicates the texture area is shrinking. The distribution of

distortions is depicted in Fig. 4.5. Texel aspect distortion stays the same, except that the space the axes are inverted compared to TSC. Λ_x increases along the X-axis, while Λ_y increases along the Y-axis. Far better property of ASC, considering the area distortion, can be verify also by comparing Fig. 4.6 and Fig. 4.3.

4.3. Outerra Spherical Cube

Outerra Spherical Cube (OSC) is an SCM projection used in the Outerra planetary 3D engine [8]. The engine has the ability to render the whole planet with a full range of detail levels, from space down to individual blades of grass, and thus requires more uniform sampling than previous schemes. Unlike the other described projections, OSC does not have a closed form for the forward transformation. Hence, a Newton's iterative method is used in algorithm 1.

```

 $x = \sin \varphi \cos \theta$ 
 $y = \sin \theta$ 
 $z = \sqrt{1 - x^2 - y^2}$ 
 $M = (1/(2\sqrt{2} - 2) - 1) = 0.207106781$ 
 $a = Mx^2y^2$ 
 $b = -M(x^2 + y^2)$ 
 $c = -z$ 
 $d = 1 + M$ 
repeat
   $F = az^4 + bz^2 + cz + d$ 
   $F' = 4az^3 + 2bz + c$ 
   $dF = \frac{F}{F'}$ 
   $z = z - dF$ 
until  $|dF| < \epsilon$ 
 $x = zx$ 
 $y = zy$ 
```

Algorithm 1: OSC forward transformation

Inverse transformation:

$$\begin{aligned}
 M &= (1/(2\sqrt{2} - 2) - 1) = 0.207106781 \\
 z &= 1 + M(1 - x^2)(1 - y^2) \\
 l &= \sqrt{x^2 + y^2 + z^2} \\
 (4.5) \quad x &= x/l \\
 y &= y/l \\
 \varphi &= \arcsin \left(\frac{x}{\cos(\arcsin(y))} \right) \\
 \theta &= \arcsin(y)
 \end{aligned}$$

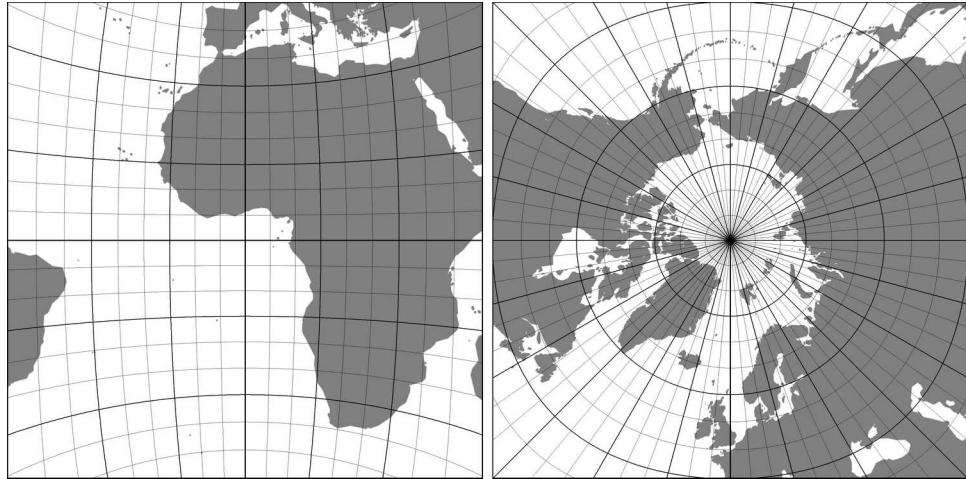


FIG. 4.7: Front and top faces of the OSC projection.

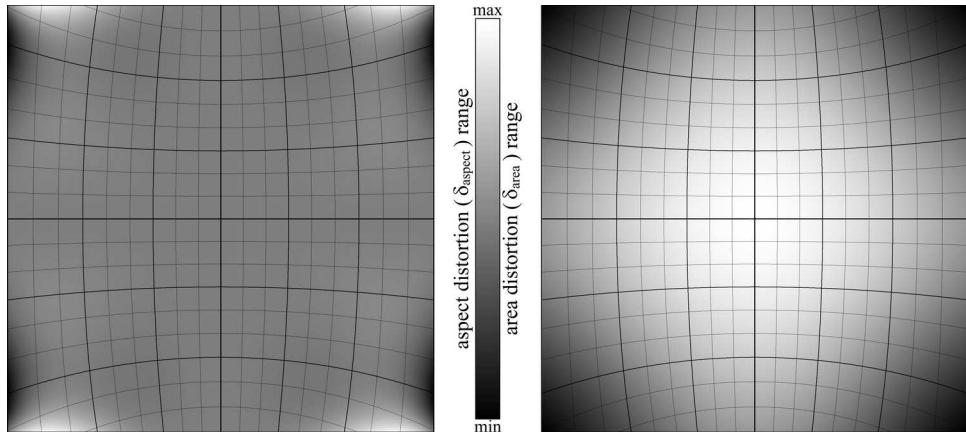


FIG. 4.8: The distribution of the aspect (left side) and area (right side) distortion over the face of the cube when OSC projection is used. The texel aspect distortion ranges from 0.934 to 1.006 ($\delta_{\text{aspect}}^{\text{max}}/\delta_{\text{aspect}}^{\text{min}} = 1.013$), while the texel area distortion ranges from 0.324 to 1.0 ($\delta_{\text{area}}^{\text{max}}/\delta_{\text{area}}^{\text{min}} = 3.088$).

OSC has the least aspect distortion of all SCM projections described in this paper. There is another approach with no aspect distortion - a conformal SCM projection proposed in [21]. However, it is based on infinite Taylor series, and as such it is less suitable for our purpose.

The OSC texel area distortion (δ_{area}) is significant and it radially increases with the distance from the center of the cube face (Fig. 4.8). It is less than in the case of TSC, but much greater than for any other SCM projection.

Although the forward transformation requires iterations, the convergence is fast. With at most five iterations, a very high precision is achieved. The maximum error for the Earth-sized sphere, imposed by transformations, is less than 1nm if double extended precision is used.

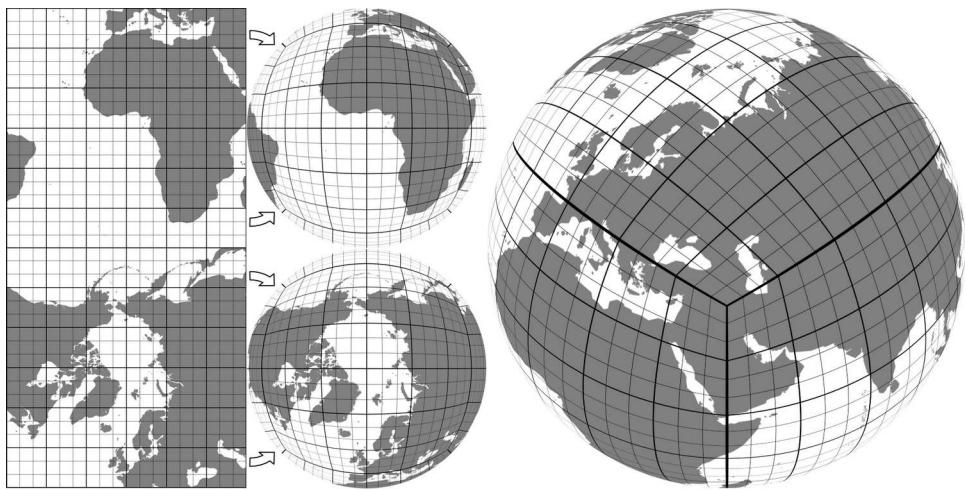


FIG. 4.9: The inverse transformation of the rectangular equidistant grid in OSC planar space and the continent coastlines to a globe surface.

4.4. COBE Quadrilateralized Spherical Cube

COBE quadrilateralized Spherical Cube (CSC) is an SCM projection based on research on the feasibility of a Quadrilateralized Spherical Cube (QLSC) Earth Data Base system, carried out in the early 1970s [3]. The purpose of the proposed projection was to minimize both area and shape distortion, and it was used, at least as it was reported, primarily in the U.S. Navy and later at NASA for the COsmic Background Explorer (COBE) project.

Forward transformation:

$$\begin{aligned}
 \tilde{x} &= \tan \varphi \\
 \tilde{y} &= \frac{\tan \theta}{\cos \varphi} \\
 x &= F(\tilde{x}, \tilde{y}) \\
 y &= F(\tilde{y}, \tilde{x})
 \end{aligned}
 \tag{4.6}$$

$$\begin{aligned}
F(x, y) = & x\gamma + x^3(1 - \gamma) \\
& + xy^2(1 - x^2) \left[\Gamma + (M - \Gamma)x^2 \right. \\
(4.7) \quad & \left. + (1 - y^2) \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} (C_{ij}x^{2i}y^{2j}) \right] \\
& + x^3(1 - x^2) \left[\Omega - (1 - x^2) \sum_{i=0}^{\infty} D_i x^{2i} \right]
\end{aligned}$$

However, QLSC has wrong forward transformation (or inverse transformation, if the notation from the original paper is used), which disqualifies it from any serious usage. That is the reason it is omitted from this paper, although it was a very important reference for all later spherical cube map studies. CSC was probably an effort to modify QLSC to be used in COBE project. As reported by Calabretta [1], the initial numeric parameters and equations derived in [3] were changed, as defined in the following formulae.

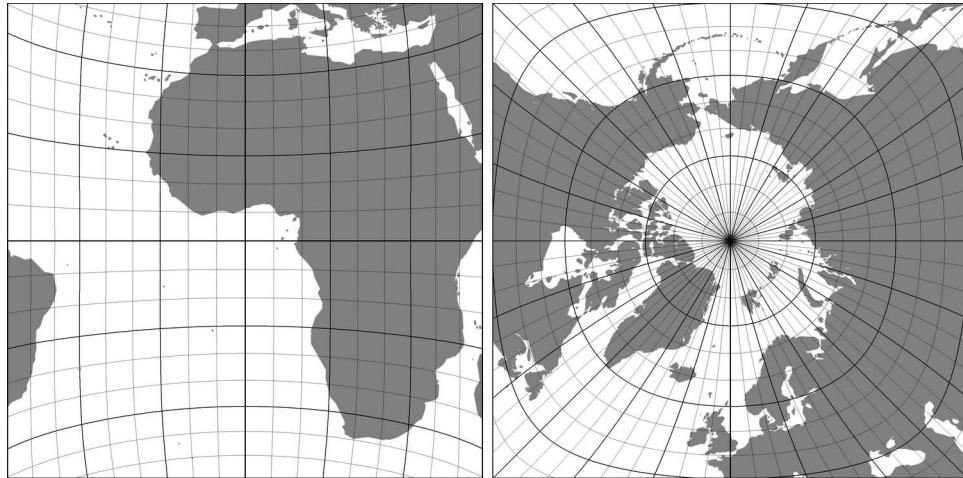


FIG. 4.10: Front and top faces of the CSC projection.

Instead of an infinite series, practical implementations use only a few terms. The following set of parameters is in use for the CSC:

$$\begin{aligned}
\gamma &= 1.37484847732 & C_{00} &= 0.141189631152 \\
M &= 0.004869491981 & C_{10} &= 0.0809701286525 \\
\Gamma &= -0.13161671474 & C_{01} &= -0.281528535557 \\
\Omega &= -0.159596235474 & C_{20} &= -0.178251207466 \\
D_0 &= 0.0759196200467 & C_{11} &= 0.15384112876 \\
D_1 &= -0.0217762490699 & C_{02} &= 0.106959469314
\end{aligned}
\tag{4.8}$$

Inverse transformation:

$$\begin{aligned}
 \tilde{x} &= I(x, y) \\
 \tilde{y} &= I(y, x) \\
 (4.9) \quad I(x, y) &= x + x(1 - x^2) \sum_{j=0}^N \sum_{i=0}^{N-j} P_{ij} x^{2i} y^{2j} \\
 \varphi &= \arctan \tilde{x} \\
 \theta &= \arctan (\tilde{y} \cdot \cos \varphi)
 \end{aligned}$$

The COBE implementation confines N to 6 and uses the following set of best-fitting values for coefficients P_{ij} :

$$\begin{aligned}
 (4.10) \quad P_{00} &= -0.27292696 & P_{12} &= -0.56800938 & P_{30} &= 0.54852384 \\
 P_{01} &= -0.02819452 & P_{13} &= 1.50880086 & P_{31} &= -1.74114454 \\
 P_{02} &= 0.27058160 & P_{14} &= -1.41601920 & P_{32} &= 0.98938102 \\
 P_{03} &= -0.60441560 & P_{15} &= 0.52032238 & P_{33} &= 0.08693841 \\
 P_{04} &= 0.93412077 & P_{20} &= -0.22797056 & P_{40} &= -0.62930065 \\
 P_{05} &= -0.63915306 & P_{21} &= 0.48051509 & P_{41} &= 1.71547508 \\
 P_{06} &= 0.14381585 & P_{22} &= 0.30803317 & P_{42} &= -0.83180469 \\
 P_{10} &= -0.07629969 & P_{23} &= -0.93678576 & P_{50} &= 0.25795794 \\
 P_{11} &= -0.01471565 & P_{24} &= 0.33887446 & P_{51} &= -0.53022337 \\
 P_{60} &= 0.02584375
 \end{aligned}$$

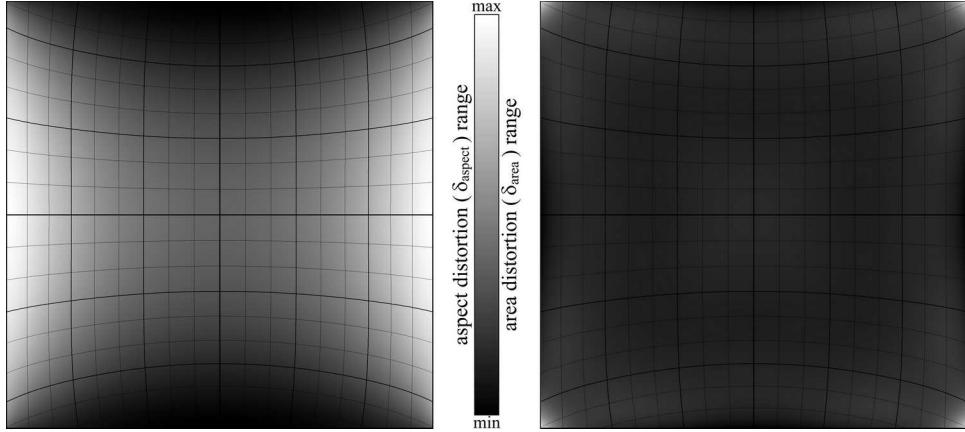


FIG. 4.11: The distribution of the aspect (left side) and area (right side) distortion over the face of the cube when CSC projection is used. The texel aspect distortion ranges from 0.65 to 1.538 ($\delta_{\text{aspect}}^{\text{max}} / \delta_{\text{aspect}}^{\text{min}} = 2.365$), while the texel area distortion ranges from 0.94 to 1.325 ($\delta_{\text{area}}^{\text{max}} / \delta_{\text{area}}^{\text{min}} = 1.41$).

CSC is an approximately equal-area projection with $\delta_{\text{area}} \in [0.94, 1.32]$. The ratio of maximum and minimum texel area distortions is the same as for ASC

($\delta_{\text{area}}^{\max}/\delta_{\text{area}}^{\min} = 1.41$), but the distribution of values is much better ($\text{RMSD}(\delta_{\text{area}}) = 0.019$). The texel aspect distortion is worse than all previously mentioned projections ($\delta_{\text{aspect}} \in [0.65, 1.54]$).

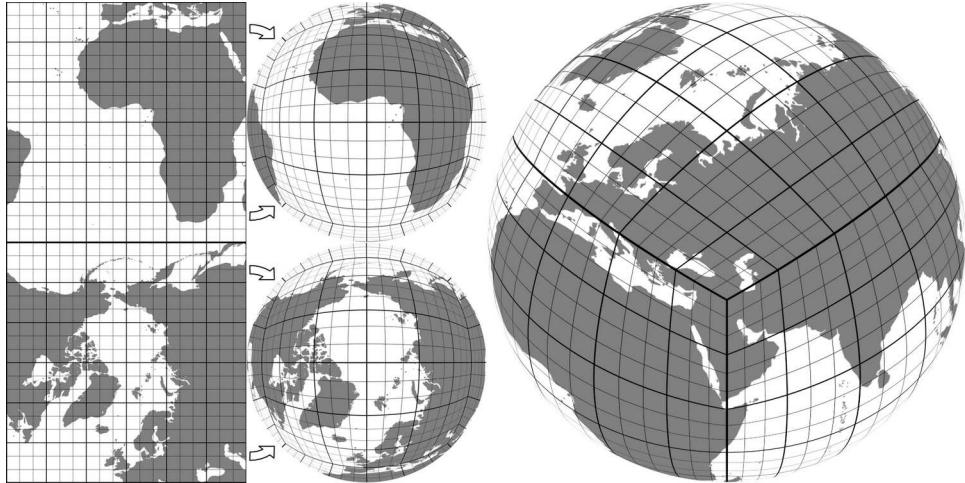


FIG. 4.12: The inverse transformation of the rectangular equidistant grid in CSC planar space and the continent coastlines to a globe surface.

Worse than the texel aspect distortion is the imprecision of the projection. The approximation imposed by using a finite number of terms and best-fitting coefficients in Eq. 4.6 and 4.9 causes a significant error in positioning. After successive inverse and forward transformations, the maximum absolute error of the position on the equivalent sphere is 1.39km. Moreover, CSC stretches an area toward the cube edges. It can be clearly seen in Fig. 4.12. The middle part of Europe almost disappeared at the edge of the cube, while Greenland is much wider than it should be. All these properties disqualify CSC for the application in any geographic information system and, hence, in SCM also.

4.5. Quadrilateralized Spherical Cube

Quadrilateralized Spherical Cube (QSC) is another SCM projection based on the work of Chan and O'Neill [3], described in Sec. 4.4. O'Neill and Laubscher [20] defined an equal-area projection to map the sphere surface to a cube face with the purpose of storing data in hierarchical structures for each cube face. In addition to being equal-area, the QSC projection was designed to limit angular distortions.

Forward transformation:

$$\begin{aligned}
 q &= \cos(\theta) \cos(\varphi) \\
 r &= \cos(\theta) \sin(\varphi) \\
 s &= \sin(\theta) \\
 \tilde{\varphi} &= \arccos(q) \\
 \tilde{\theta} &= \arctan(s, r) \\
 (4.11) \quad \mu &= \arctan\left(\frac{12}{\pi}\right) \cdot (\tilde{\theta} + \arccos(\sin(\tilde{\theta}) \cos(\frac{\pi}{4})) - \frac{\pi}{2}) \\
 \nu &= \arctan\left(\sqrt{\frac{1 - \cos(\tilde{\varphi})}{\cos^2(\mu) \cdot (1 - \cos(\arctan(1/\cos(\tilde{\theta}))))}}\right) \\
 x &= \tan(\nu) \cos(\mu) \\
 y &= \tan(\nu) \sin(\mu)
 \end{aligned}$$

Inverse transformation:

$$\begin{aligned}
 \nu &= \arctan(\sqrt{x^2 + y^2}) \\
 \mu &= \arctan\left(\frac{y}{x}\right) \\
 t &= \frac{\pi}{12} \tan(\mu) \\
 \tilde{\theta} &= \arctan\left(\frac{\sin(t)}{\cos(t) - 1/\sqrt{2}}\right) \\
 (4.12) \quad \tilde{\varphi} &= \arccos(1 - \cos^2(\mu) \tan^2(\nu)(1 - \cos(\arctan(1/\cos(\tilde{\theta}))))) \\
 q &= \cos(\tilde{\varphi}) \\
 s &= \sqrt{1 - q^2} \sin(\tilde{\theta}) \\
 r &= \sqrt{1 - q^2 - s^2} \\
 \theta &= \arccos(-s) - \frac{\pi}{2} \\
 \varphi &= \arctan\left(\frac{r}{q}\right)
 \end{aligned}$$

The formulae for forward and inverse transformations, given above, apply to one-quarter of the front cube face; the other three-quarters are handled by rotating this definition. This is done by first determining the interval of $\tilde{\theta}$, which defines the quarter, then shifting $\tilde{\theta}$ to the interval of definition $[-\frac{\pi}{4}, \frac{\pi}{4}]$ by adding or subtracting a multiple of $\frac{\pi}{2}$, then computing μ as described, and finally shifting μ back to the original quarter by again adding or subtracting a multiple of $\frac{\pi}{2}$. Furthermore, other cube faces than the front cube face are handled by adapting the computation of $\tilde{\theta}$, e.g. $\tilde{\theta} = \arctan(s, -q)$ for the appropriate cube face.

As can be seen in Fig. 4.13, QSC suffers from significant shape distortion. Furthermore, there are discontinuities at the $x = |y|$ directions (diagonals of the cube

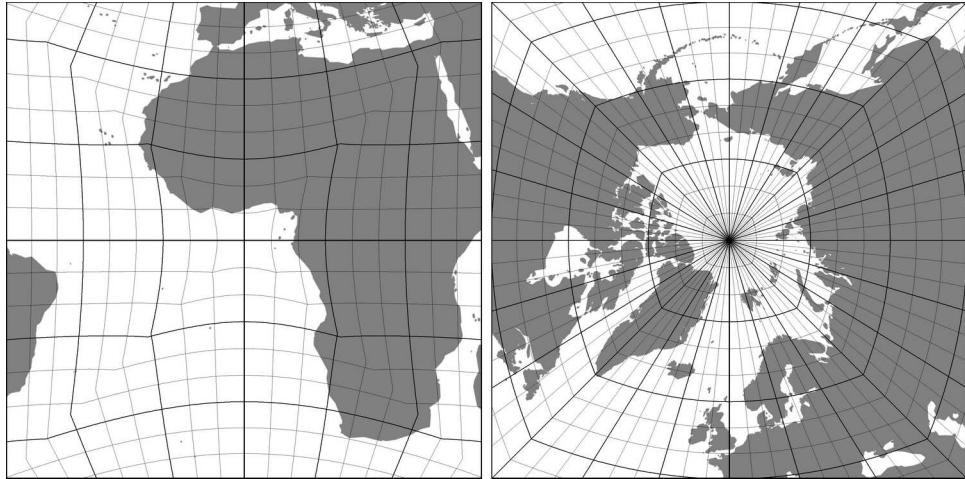
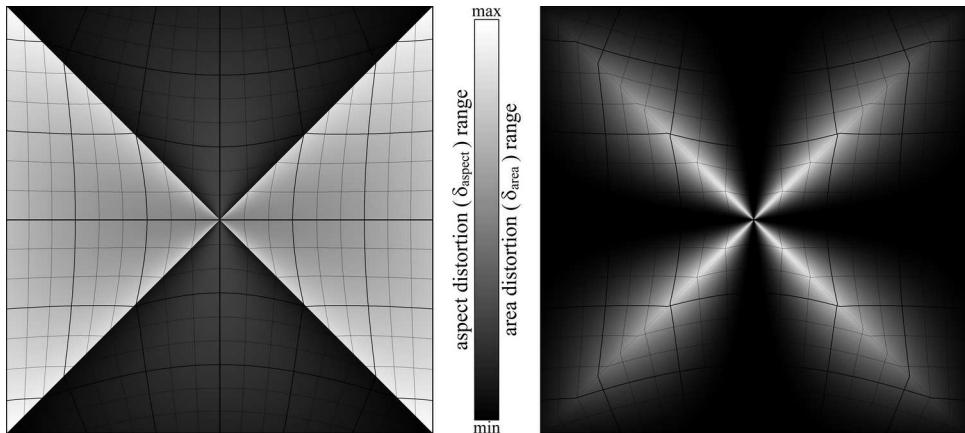


FIG. 4.13: Front and top faces of the QSC projection.

FIG. 4.14: The distribution of the aspect (left side) and area (right side) distortion over the face of the cube when QSC projection is used. The texel aspect distortion ranges from 0.649 to 1.539 ($\delta_{\text{aspect}}^{\text{max}}/\delta_{\text{aspect}}^{\text{min}} = 2.37$), while the texel area distortion ranges from 0.89 to 0.93 ($\delta_{\text{area}}^{\text{max}}/\delta_{\text{area}}^{\text{min}} = 1.042$).

faces). These discontinuities change the direction of aspect distortion (which is also at its maximum), cause severe additional texture distortion if intersecting triangles of the underlying mesh and also slightly disturb the equal-area property of the projection. The left side of Fig. 4.16 depicts a distortion caused by intersecting triangles, while the underlying mesh is shown on the right side of the Fig. 4.16. This issue can be solved by splitting the cube faces into four triangular regions, using a very fine tessellation (a pixel-sized triangles) or ray casting rendering (per

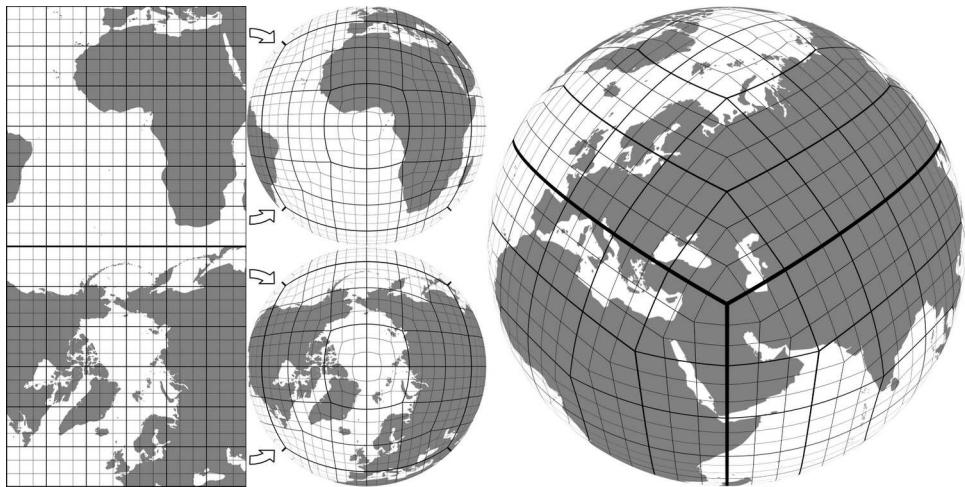


FIG. 4.15: The inverse transformation of the rectangular equidistant grid in QSC planar space and the continent coastlines to a globe surface.

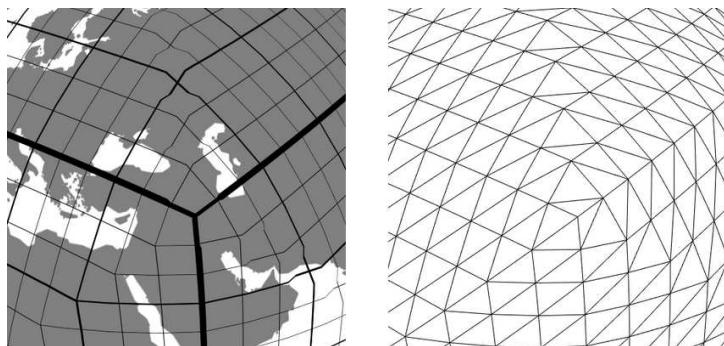


FIG. 4.16: Discontinuities at the cube face diagonals can cause severe texture distortion if they intersect triangles of the textured mesh. Diagonals toward the pole and India ripple the grid, while the diagonal toward the center of Africa does not, since the line of discontinuity is aligned with triangles' edges.

pixel texture sampling).

4.6. Cartesian Spherical Cube

Cartesian Spherical Cube (KSC) emerged from mapping a square to a circle [18] and its generalization to mapping a cube to a sphere [17], proposed by Philip Nowell. The proposed mapping is actually the inverse transformation in a closed form with coordinates defined in Cartesian coordinate system (Eq. 4.15). The initial in-

tent was not to provide a cartographic mapping, but just a sphere parametrization example posted on the Web blog [17]. However, some nice properties of this transformation yielded an implementation of the forward transformation [25] five years later. Prior to the forward transformation (Eq. 4.14), the polar coordinates have to be transformed to the Cartesian coordinates (Eq. 4.13) and the cube face has to be determined (the third row in the Tab. 4.1) according to one-sixth the maximum of all three Cartesian coordinates (the first row in the Tab. 4.1) and its sign (the second row in the Tab. 4.1). Depending on the cube face, input variables for the transformation have to be rearranged (the fourth row in the Tab. 4.1), so that each face can be transformed to the front face of the cube.

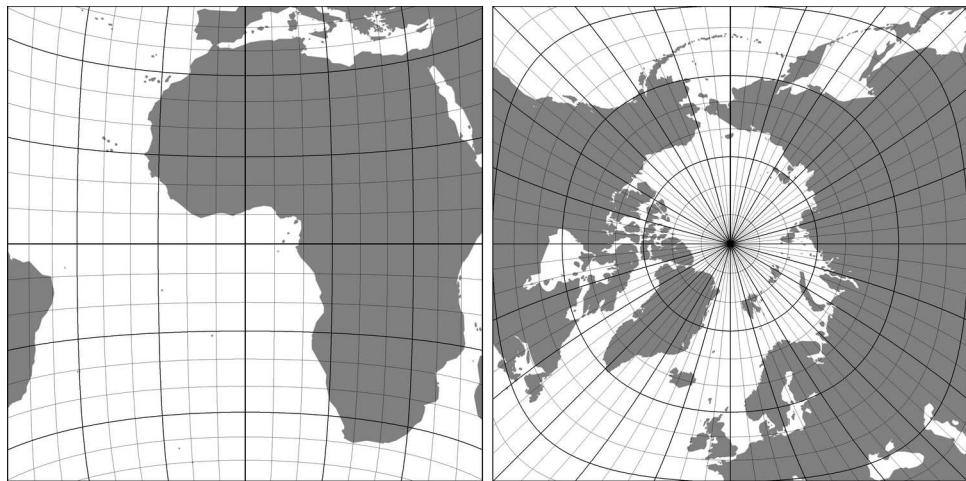


FIG. 4.17: Front and top faces of the KSC projection.

$(x \geq y) \wedge (x \geq z)$		$(y \geq x) \wedge (y \geq z)$		$(z \geq x) \wedge (z \geq y)$	
$x > 0$	$x < 0$	$y > 0$	$y < 0$	$z > 0$	$z < 0$
Face 1 (right)	Face 3 (left)	Face 4 (top)	Face 5 (bottom)	Face 0 (front)	Face 2 (back)
$x' = -z$	$x' = z$	$y' = -z$	$y' = z$		$x' = -x$
$z' = x$	$z' = x$	$z' = y$	$z' = y$		

Table 4.1: Face selection and mapping to a front face before KSC forward transformation.

Forward transformation:

$$(4.13) \quad \begin{aligned} \chi &= \cos \theta \sin \varphi \\ \psi &= \sin \theta \\ \zeta &= \cos \theta \cos \varphi \end{aligned}$$

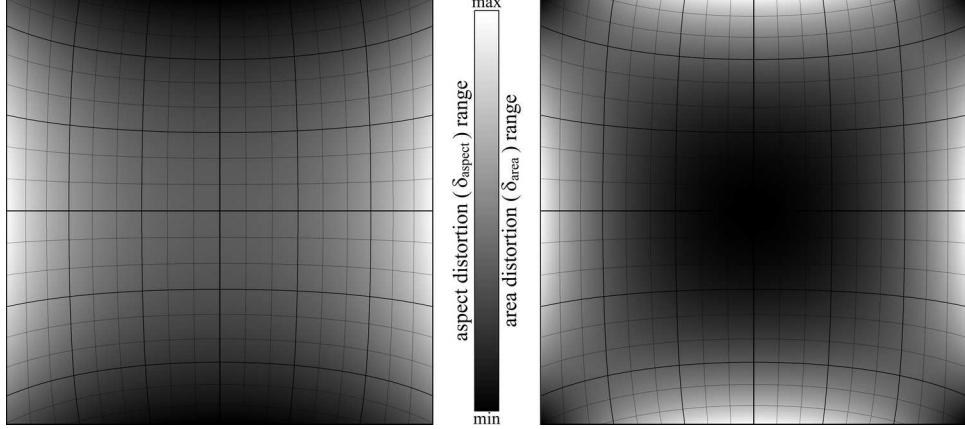


FIG. 4.18: The distribution of the texel aspect (left side) and area (right side) distortion over the face of the cube when KSC projection is used. The texel aspect distortion ranges from 0.577 to 1.731 ($\delta_{\text{aspect}}^{\max}/\delta_{\text{aspect}}^{\min} = 3.0$), while the texel area distortion ranges from 1.0 to 1.155 ($\delta_{\text{area}}^{\max}/\delta_{\text{area}}^{\min} = 1.155$).

$$\begin{aligned}
 \xi &= -\sqrt{(2\psi^2 - 2\chi^2 - 3)^2 - 24\chi^2} \\
 x &= \text{sign}(\chi) \cdot \min \left(\sqrt{\frac{\max(\xi + 2\chi^2 - 2\psi^2 + 3, 0)}{2}}, 1 \right) \\
 (4.14) \quad y &= \text{sign}(\psi) \cdot \min \left(\sqrt{\frac{\max(\xi - 2\chi^2 + 2\psi^2 + 3, 0)}{2}}, 1 \right) \\
 z &= \text{sign}(\zeta)
 \end{aligned}$$

Inverse transformation:

$$\begin{aligned}
 (4.15) \quad x' &= x \cdot \sqrt{\max(1 - \frac{1}{2}y^2 - \frac{1}{2}z^2 + \frac{1}{3}y^2z^2, 0)} \\
 y' &= y \cdot \sqrt{\max(1 - \frac{1}{2}x^2 - \frac{1}{2}z^2 + \frac{1}{3}x^2z^2, 0)} \\
 z' &= z \cdot \sqrt{\max(1 - \frac{1}{2}x^2 - \frac{1}{2}y^2 + \frac{1}{3}x^2y^2, 0)}
 \end{aligned}$$

$$\begin{aligned}
 (4.16) \quad \varphi &= \arctan \frac{x'}{z'} \\
 \theta &= \arcsin y'
 \end{aligned}$$

KSC is an approximately equal-area SCM projection (15% deviation), with a severe texel aspect distortion. It has worse aspect distortion than any other previously mentioned SCM. Another unusual property of the KSC is both shrinking and

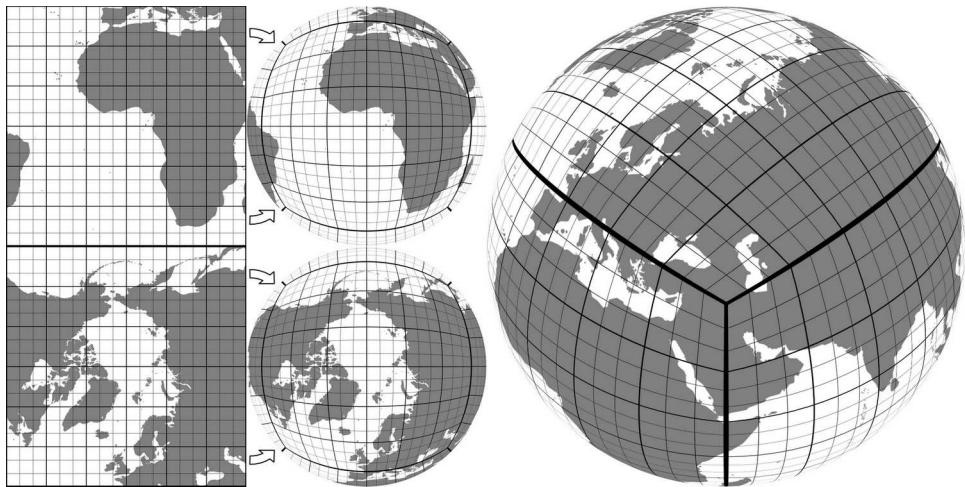


FIG. 4.19: The inverse transformation of the rectangular equidistant grid in KSC planar space and the continent coastlines to a globe surface.

stretching of the texels at the same time along different axes, while moving from the center of the cube face toward the middle of the edges. The texel aspect is constant at the diagonals.

4.7. Hierarchical Equal Area isoLatitude Pixelization

Hierarchical Equal Area isoLatitude Pixelization (HEALPix) [6] is a class of spherical projections with a property of distributing $12N^2$ points as uniformly as possible over the surface of the unit sphere. These hybrid projections combine the Lambert cylindrical equal-area projection, for the equatorial region with the interrupted Collignon projection for the polar regions. This infinite class of projections is parameterized by N_θ and N_ϕ [6] (often referred to as K and H, respectively [2]). N_θ is the number of base-resolution pixel layers between the north and south poles and N_ϕ is the multiplicity of the meridional cuts, or the number of equatorial or circumpolar base-resolution pixels. In this paper, we discuss only the HEALPix projection with $N_\theta = 3$ and $N_\phi = 4$ (Fig. 4.20), since it is the only projection of the whole class that can be rearranged to a cube-based hexahedral projection.

The equatorial and the polar regions meet at latitude $\tilde{\theta}$. This particular latitude can be calculated based on the fact that the polar region, as a part of an equal-area projection, needs to be one-sixth of the total area, and that the area of the spherical cap can be calculated as $P = 2\pi(1 - \sin \theta)$ [2]. Hence, $\tilde{\theta} = \arcsin(2/3) \approx 41.81^\circ$.

Since the projection differs for the equatorial and polar regions, we will provide forward and inverse transformations separately. The forward transformation of the

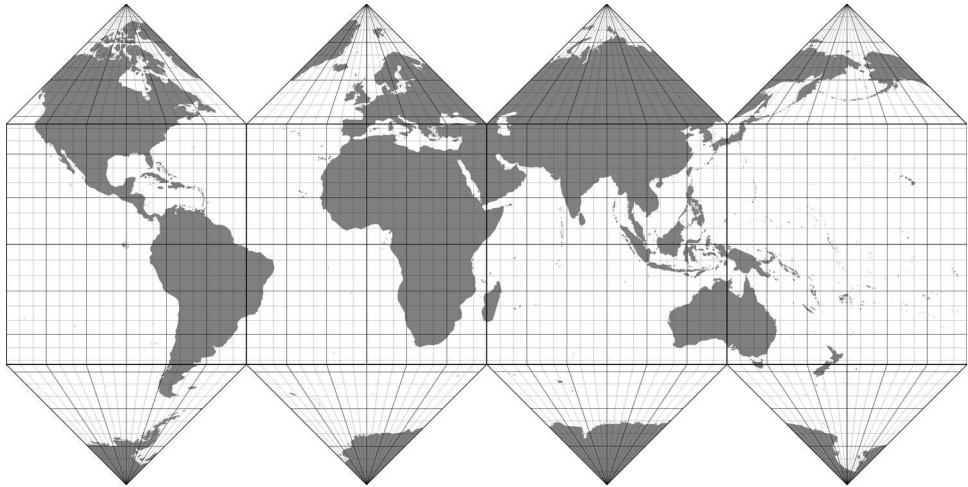


FIG. 4.20: The world map based on HEALPix projection with $N_\theta = 3$ and $N_\phi = 4$.

equatorial region ($|\theta| < \tilde{\theta}$) is:

$$(4.17) \quad \begin{aligned} x &= \varphi \cdot \frac{4}{\pi} \\ y &= \frac{3}{2} \cdot \sin \theta \end{aligned}$$

Inverse transformation for equatorial region:

$$(4.18) \quad \begin{aligned} \varphi &= x \cdot \frac{\pi}{4} \\ \theta &= \arcsin \frac{2y}{3} \end{aligned}$$

As it can be seen in the Fig. 4.22, HEALPix is an exact equal-area projection for the equatorial region. Also, the texel aspect distortion (δ_{aspect}) is less than for any other SCM projection, except for OSC.

HEALPix projections are not cube-based, but the four triangles of the interrupted Collignon projection for the polar regions can be rearranged and grouped to form a face of the cube. The process of rearrangement after the forward transformation is summarized in Tab. 4.2. This process should be reversed in the inverse transformation.

Forward transformation for polar regions:

$$(4.19) \quad \begin{aligned} \sigma &= \sqrt{3(1 - |\sin \theta|)} \\ x &= \sigma \varphi \cdot \frac{4}{\pi} \\ y &= 1 - \sigma \end{aligned}$$

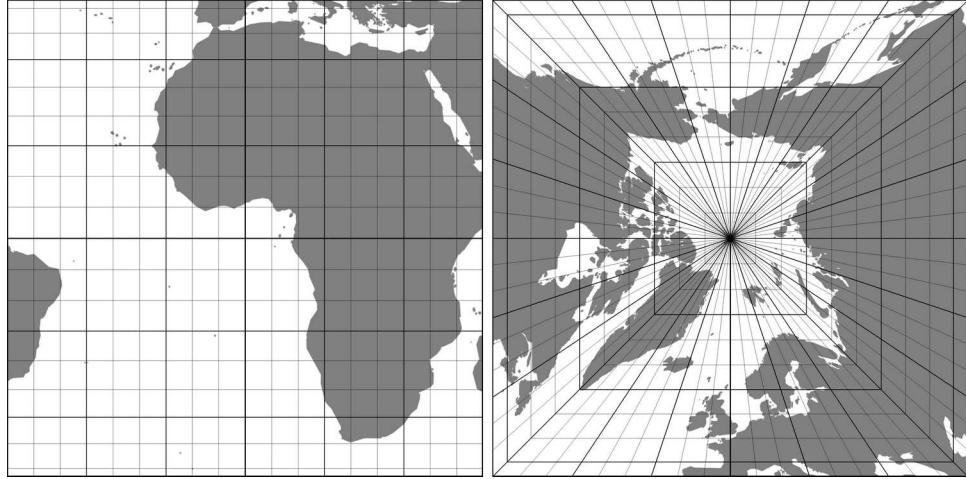


FIG. 4.21: Front and top faces of the HEALPix ($N_\theta = 3$ and $N_\phi = 4$) projection.

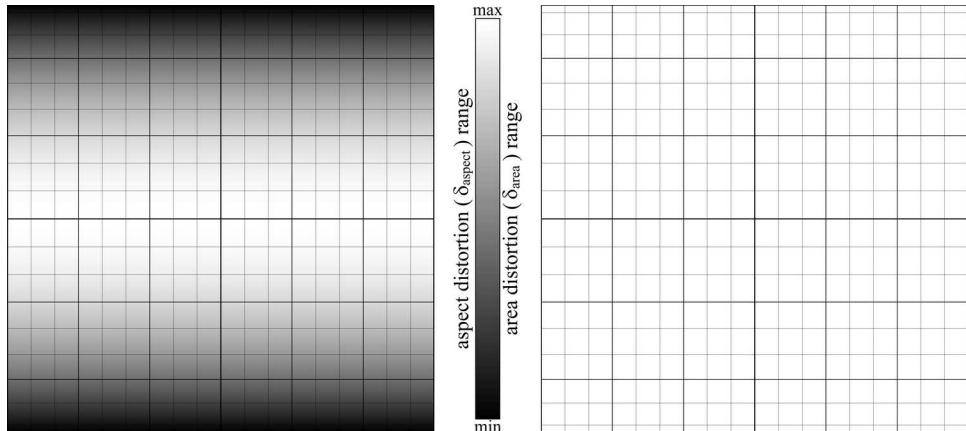


FIG. 4.22: . The distribution of the aspect (left side) and area (right side) distortion over the equatorial face of the cube when HEALPix ($N_\theta = 3$ and $N_\phi = 4$) projection is used. The texel aspect distortion ranges from 0.654 to 1.178 ($\delta_{\text{aspect}}^{\text{max}}/\delta_{\text{aspect}}^{\text{min}} = 1.8$). There is no area distortion.

Inverse transformation for polar regions:

$$\begin{aligned}
 \sigma &= 1 - y \\
 \varphi &= \frac{x}{\sigma} \cdot \frac{\pi}{4} \\
 (4.20) \quad \theta &= \pm \arcsin \left(1 - \frac{\sigma^2}{3} \right)
 \end{aligned}$$

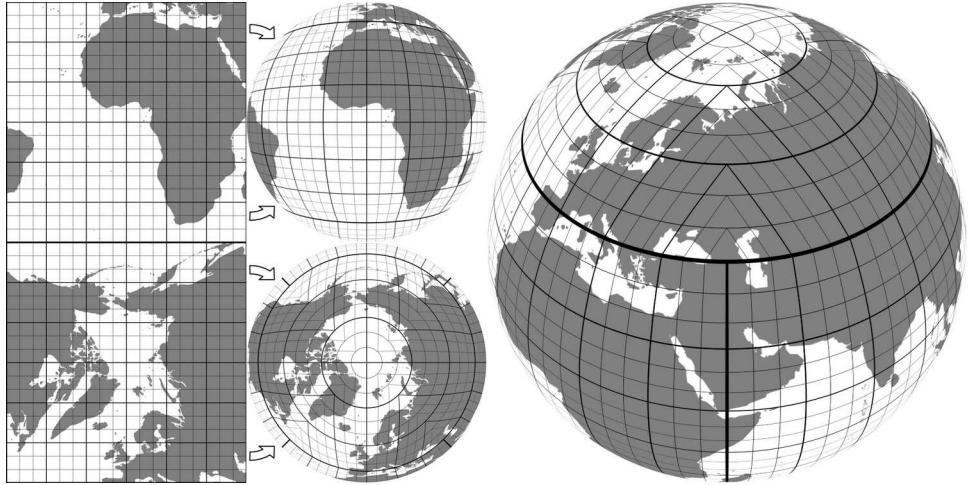


FIG. 4.23: The inverse transformation of the rectangular equidistant grid in HEALPix ($N_\theta = 3$ and $N_\phi = 4$) planar space and the continent coastlines to a globe surface.

φ	$[-\frac{\pi}{4}, \frac{\pi}{4}]$	$[\frac{\pi}{4}, \frac{3\pi}{4}]$	$[\frac{3\pi}{4}, \frac{5\pi}{4}]$	$[\frac{5\pi}{4}, \frac{7\pi}{4}]$
$\theta > \tilde{\theta}$	$y = y-1$ $x = 1-y$ $y = z$	$z = x$ $y = 1-y$	$x = -x$ $y = -y$	$z = x$ $x = y-1$ $y = -z$
$\theta < -\tilde{\theta}$	$y = 1-y$	$z = x$ $x = 1-y$ $y = -z$	$x = -x$ $y = y-1$	$z = x$ $x = y-1$ $y = z$

Table 4.2: Rearrangement of interrupted Collignon projection's triangles into a quad after the forward transformation.

Subface	$y \leq x $	$-x \leq y \leq x$	$y \geq x $	$-y \leq x \leq y$
North polar face	$y = 1+y$	$z = y$ $y = 1-x$ $x = z$	$x = -x$ $y = 1-y$	$z = y$ $y = 1+x$ $x = -z$
South polar face	$x = -x$ $y = 1+y$ $x = z$	$z = y$ $y = 1+x$	$y = 1-y$	$z = y$ $y = 1-x$ $x = -z$

Table 4.3: Splitting the polar quads into interrupted Collignon projection's triangles before the inverse transformation.

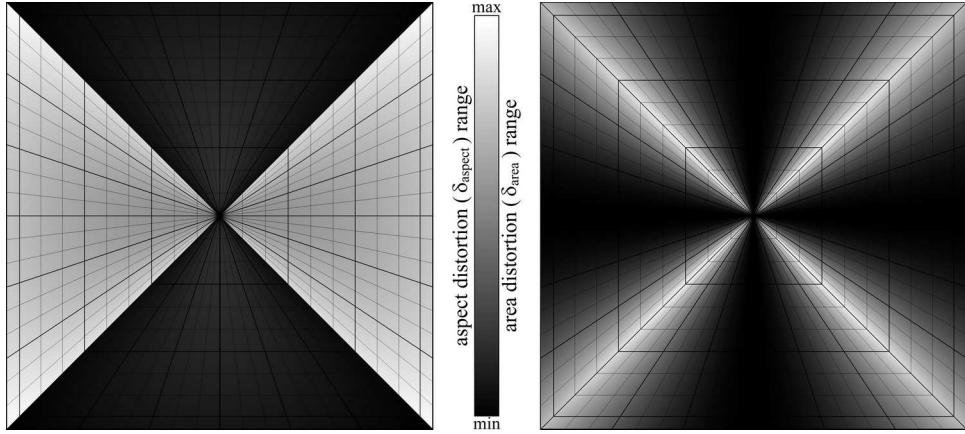


FIG. 4.24: The distribution of the aspect (left side) and area (right side) distortion over the polar face of the cube when HEALPix projection is used. The texel aspect distortion ranges from 0.548 to 1.826 ($\delta_{\text{aspect}}^{\text{max}}/\delta_{\text{aspect}}^{\text{min}} = 3.33$), while the texel area distortion ranges from 1.0 to 1.27 ($\delta_{\text{area}}^{\text{max}}/\delta_{\text{area}}^{\text{min}} = 1.27$).

Texel aspect distortion is very high in the polar regions (left side of Fig. 4.24). It has the highest value of all described projections, and it can be clearly seen on the graticule and continents shape (right side of the Fig. 4.21). Like in QSC, there are discontinuities at $x = |y|$ directions on the polar faces of the cube, with even higher magnitude. These discontinuities cause the same problems as with QSC, like further increase of the aspect distortion (it is at maximum and change direction at the line of discontinuity), texture rippling over triangles that intersect discontinuity and also disturbing the equal-area property (right side of the Fig. 4.24).

5. Spherical Cube Map Projection Comparison

In this section, we summarize the characteristics of the projections, covered in the previous sections, and present a side-by-side comparison according to the tests described in Sec. 3.

Although precision was not the main criteria used in SCM projection evaluation, poor precision can certainly limit a projections field of application. For example, precision is very important in location services, cadastral surveys and geographic information systems. Almost all SCM projections maintained good precision, with the error introduced by an inverse transformation followed by a forward transformation less than or equal to $1\mu\text{m}$ for an Earth-sized planet. The only exception is CSC (and also QLSC, but this projection is not covered in the paper). Because of its imprecision and tendency to stretch the surface toward the edges of the cube, CSC cannot be used for Earth mapping.

Another property used in the evaluation process is the execution time of both forward and inverse transformations. Execution time significantly depends on implementation optimization, compiler, CPU architecture, working frequency, caching scheme, etc. Therefore, Fig. 5.1 depicts normalized values. The normalization is done using the shortest execution time. The tests were executed on Intel Core i7-4700HQ CPU using the Microsoft Visual Studio 2013 C++ compiler on the Microsoft Windows 8.1 operating system. As can be seen in Fig. 5.1, HEALPix is the fastest projection, while OSC and QSC are the slowest ones. The OSC forward transformation, due to its iterative nature, has a relatively long execution time, but even so it has approximately the same speed as QSC. On the other hand, the inverse transformation of OSC is much faster than QSC. KSC has approximately the same execution time for both transformations. Generally, excluding HEALPix, all other SCM projections have the same order of magnitude execution time.

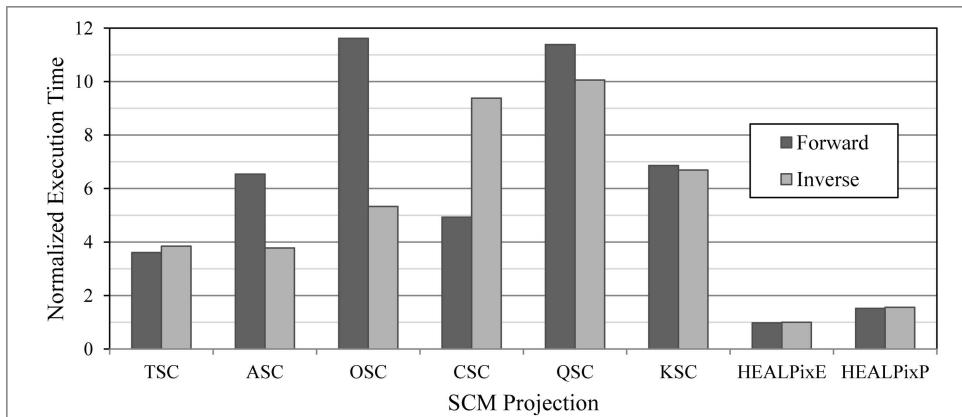


FIG. 5.1: Forward and inverse transformation execution time comparison. Values are normalized using HEALPix equatorial inverse transformation time.

The most important property used in evaluating the quality of SCM projections is distortion. Table 5.1 gives a comparative review of relevant distortion parameters: minimum (*min*) and maximum (*max*) values, maximum-to-minimum ratio (*ratio*) and root-mean-square deviation (*RMSD*) of both texel aspect and area distortion. HEALPix for the polar regions and KSC have the worst aspect distortion. QSC has a significant texel aspect distortion, while TSC and ASC are slightly better. HEALPix and QSC are actually not hexahedral projection. In order to reduce the effects of discontinuities, faces of the cube have to be divided into, at least, four triangular areas with edges aligned with discontinuities. OSC is the only SCM projection that eliminates aspect distortion.

TSC has the worst texel area distortion. OSC is better than TSC, but still has significant area distortion at the corners of the cube faces. ASC has a relatively low area distortion, while KSC can be considered approximately equal-area projection. QSC is classified as an equal-area projection, although the discontinuities slightly

disturb that property. Furthermore, in the tests of this property, we have compared texels all over the cube face with the texel in its center. However, the central texel in the QSC projection crosses both lines of discontinuities, hence its size is bigger than any other texel in the map. That is the reason the maximal value of the QSC area distortion is less than one. The even number of texels is assumed along each axis, which eliminates the extreme value and gives a little better overall equal-area property of QSC. HEALPix for the equatorial region is an exact equal-area projection, while the polar regions suffer from discontinuities the same way as QSC, but even with a higher magnitude.

Projection	δ_{aspect}				δ_{area}			
	min	max	ratio	RMSD	min	max	ratio	RMSD
TSC	0.707	1.414	2.000	0.155	0.222	1.000	4.500	0.506
ASC	0.707	1.414	2.000	0.146	0.707	1.000	1.414	0.153
OSC	0.994	1.006	1.013	0.001	0.324	1.000	3.088	0.280
CSC	0.650	1.538	2.365	0.218	0.940	1.325	1.410	0.019
QSC	0.650	1.539	2.369	0.271	0.894	0.931	1.042	0.099
KSC	0.577	1.732	3.000	0.227	1.000	1.155	1.155	0.063
HEALPix ^E	0.654	1.178	1.800	0.156	1.000	1.000	1.000	0.000
HEALPix ^P	0.548	1.826	3.334	0.437	1.000	1.272	1.272	0.108

Table 5.1: Comparative review of SCM projections relevant distortion parameters: minimum (*min*) and maximum (*max*) values, maximum-to-minimum ratio (*ratio*) and root-mean-square deviation (*RMSD*) of both texel aspect and area distortion.

Even though a tabular review is useful for comparing values, a visual representation is usually more convincing. Fig. 6.1 gives a side-by-side comparison of all evaluated SCM projections. A comparison is done through topographic view with the graticule, texel aspect distortion and texel area distortion distribution over the face of the cube. Since projections may differ in the equatorial and polar regions, both equatorial (front) and polar (top) faces are provided.

Fig. 6.2 compares the inverse transformation effects by reprojecting regular grids from the projection planes back to the spherical surface. The figure reveals how the grid is distorted and also the issues with CSC projection. It can be clearly seen that the shapes of the continents are incorrect in the case of CSC. CSC was used for mapping cosmic background radiation, where the equal-area property was important in representing its density, while the distortion effects were of secondary importance if they were relevant at all or even noticed.

6. Conclusion

A spherical surface cannot be mapped to a plane without distortion. If a projection preserves shapes, it does not preserve area and vice versa. The choice of map projection must therefore always consider the requirements of the application area.

For planet-sized terrain rendering, projections from the sphere onto the six faces of a cube are of particular interest, since the rectangular cube faces allow use of existing file formats for image and data storage, textures with mipmap and anisotropic filtering capabilities as are typical in graphics pipelines, and quadtree hierarchies and clipmaps are commonly used for level-of-detail purposes.

The rendering stage, in particular the texture sampling stage in modern graphics pipelines, dictates the quality of the rendering result of such applications. We, therefore, derive the following quality criteria for the evaluation of map projections: texel aspect distortions, texel area distortions, and efficiency of transformations required for texture sampling.

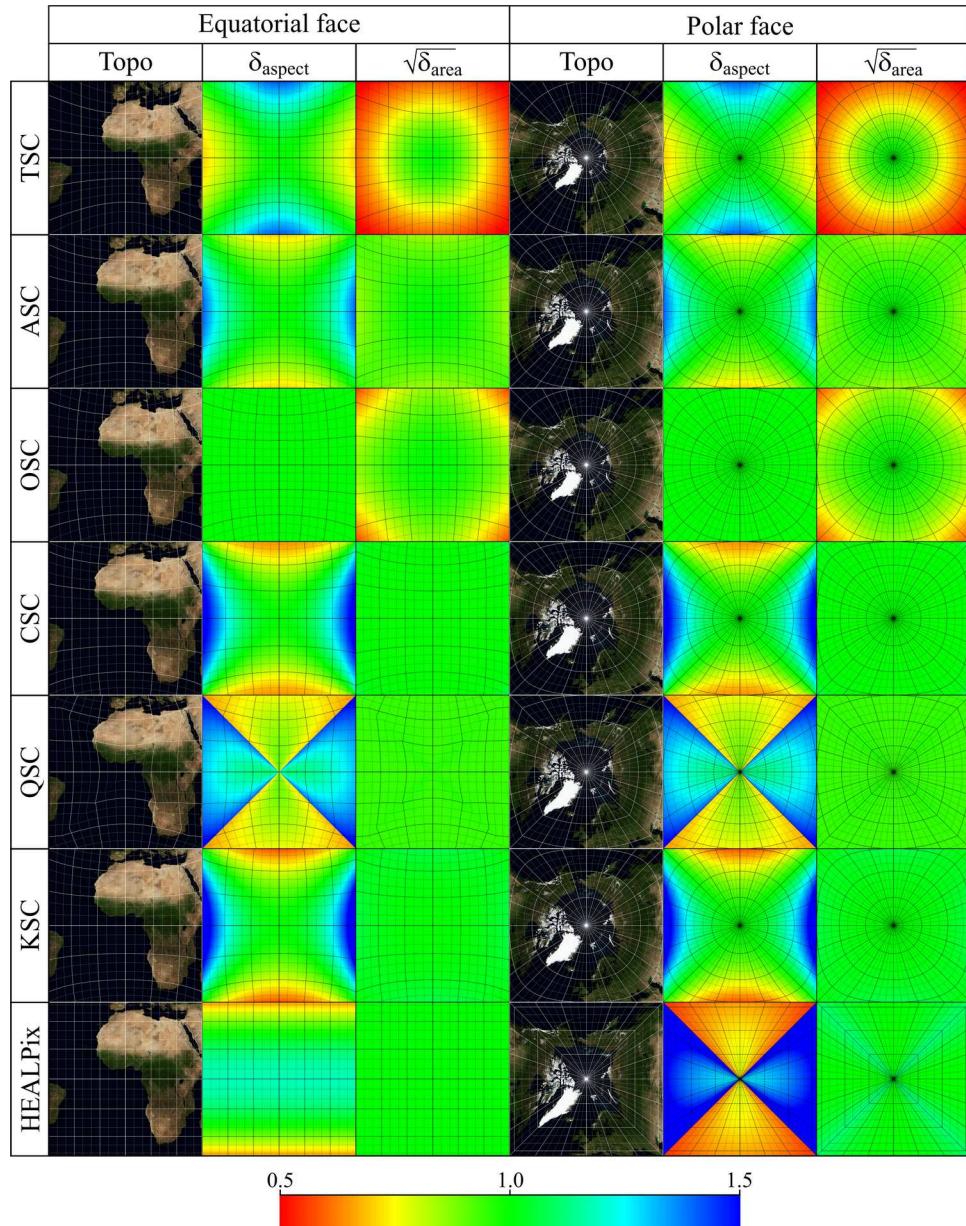
Both texel aspect and area distortion increase texture size required for the certain level of fidelity, while texel aspect distortion also spends a certain amount of the hardware supported anisotropy range. The efficiency of the transformations directly dictates the time needed for the data preparation and the rendering itself.

Using these criteria, we evaluated a comprehensive list of suitable SCM projections. Each projection has its advantages and disadvantages. A few projections are clearly unsuitable for the task; QLSC because of the wrong forward transformation and CSC because of imprecision and distortion that cannot be corrected by texture filtering. Among other presented projections, both QSC and HEALPix are actually not hexahedral and they introduce discontinuities if they are treated as such. Also, both QSC and HEALPix introduce high texel aspect distortion.

Comparing all presented SCM projections, ASC is probably the best choice, combining easy implementation, relatively fast transformation and moderate distortion. ASC requires the texture storage space twice the size of the nominal value. OSC enables sharper rendering than any other SCM, since the available anisotropy filtering range is not spent on the aspect distortion. However, OSC may require bigger texture storage space (about 54% more than ASC, if the same texture levels blending scheme is used). Also, OSC has slower forward transformation than ASC.

KSC requires approximately the same texture size as OSC, but results in the blurrier rendering of flat, nearly horizontal surfaces, because of spending a significant range of available anisotropic filtering for correcting aspect distortion. TSC is the worst choice considering required texture size (2.25 times more than ASC). Although the texel aspect distortion of the TSC is the same as of ASC, the texel area distortion is far worse than any other SCM projection.

Among the vast number of known projections (and the infinite number of as yet unknown projections), there are certainly more that can be applied to map a sphere to cube faces. Future work will, therefore, include the search for more projections, and their evaluation. Since the particular needs of the planet-sized terrain rendering were not considered when constructing most known map projections, it is possible that a projection specifically created for this task could be superior to the projections evaluated in this paper.



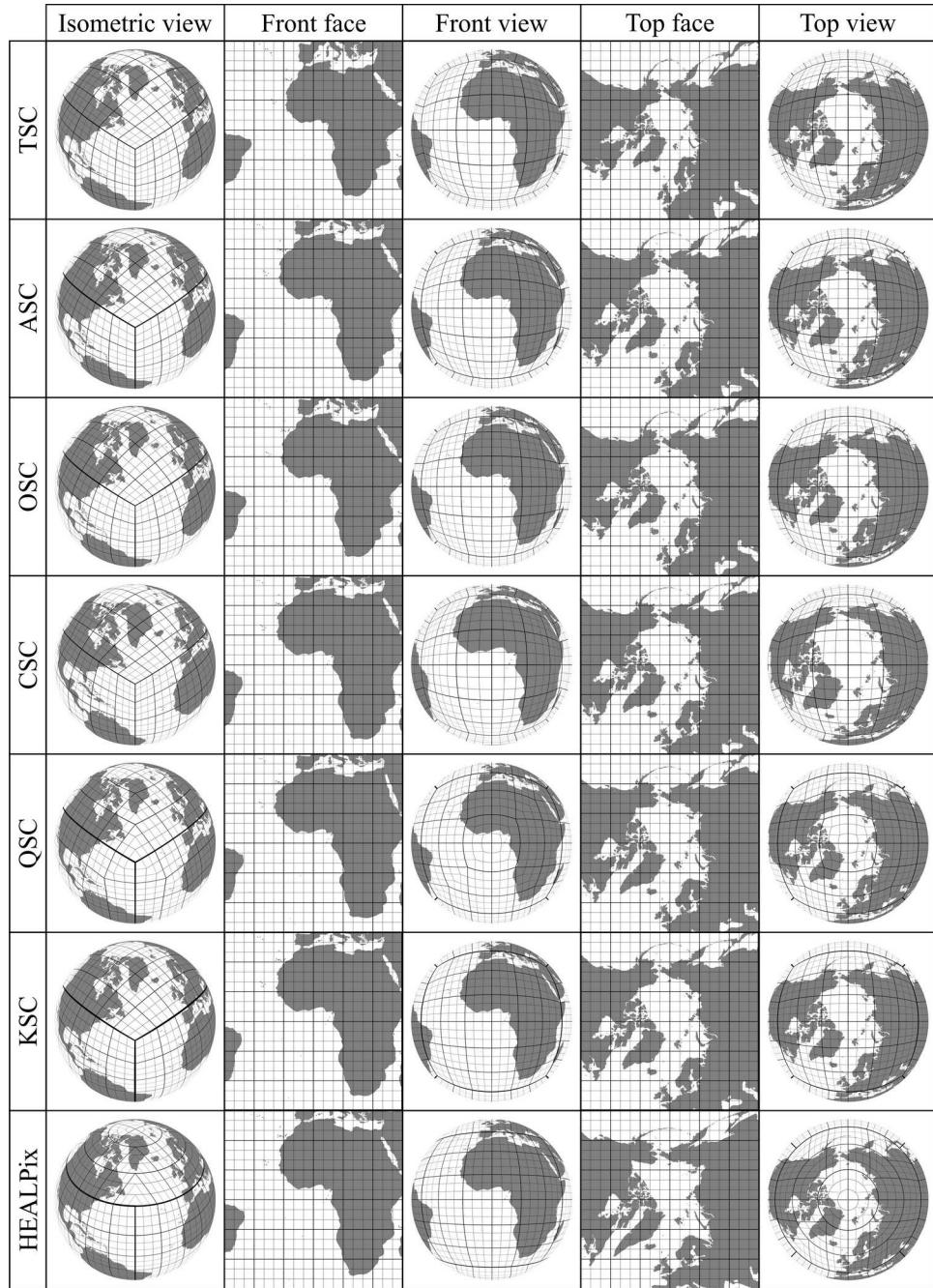


FIG. 6.2: Side-by-side comparison of the SCM inverse transformations effects. The effects are visualized by reprojecting regular grids from the projection planes back to the spherical surface.

REFERENCES

1. M. CALABRETTA AND E. GREISEN, *Representations of celestial coordinates in FITS*, Astronomy & Astrophysics, 395 (2002), pp. 1077–1122.
2. M. R. CALABRETTA AND B. F. ROUKEMA, *Mapping on the healpix grid*, Monthly Notices of the Royal Astronomical Society, 381 (2007), pp. 865–872.
3. F. CHAN AND E. O’NEILL, *Feasibility study of a quadrilateralized spherical cube earth data base*, Tech. Report EPRF 2-75 (CSC), Environmental Prediction Research Facility, Apr. 1975.
4. A. M. DIMITRIJEVIĆ AND D. D. RANČIĆ, *Ellipsoidal Clipmaps – A Planet-Sized Terrain Rendering Algorithm*, Computers & Graphics, (2015), pp. 43–61.
5. GOOGLE, *Google Earth*. <http://www.google.com/earth/index.html>, 2005. Accessed 2014-01-15.
6. K. M. GÓRSKI, E. HIVON, A. J. BANDY, B. D. WANDELT, F. K. HANSEN, M. REINECKE, AND M. BARTELmann, *HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere*, The Astrophysical Journal, 622 (2005), pp. 759–771.
7. INTEL CORPORATION, *Intel® 64 and ia-32 architectures software developer’s manual, vol.1: Basic architecture*. <http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-1-manual.pdf>, 2014.
8. B. KEMEN AND L. HRABCÁK, *Outerra*. <http://www.outerra.com>, 2014.
9. J. KESSENICH, *The OpenGL shading language, language version: 4.50*. <https://www.opengl.org/registry/doc/GLSLangSpec.4.50.pdf>, Jenuary 2015.
10. R. KOOIMA, J. LEIGH, A. JOHNSON, D. ROBERTS, M. SUBBARAO, AND T. A. DEFANTI, *Planetary-scale terrain composition*, IEEE Trans. Visualization and Computer Graphics, 15 (2009), pp. 719–733.
11. M. LAMBERS AND A. KOLB, *Ellipsoidal cube maps for accurate rendering of planetary-scale terrain data*, in Proc. Pacific Graphics (Short Papers), Sept. 2012, pp. 5–10.
12. R. LERBOUR, *Adaptive streaming and rendering of large terrains*, PhD thesis, Université de Rennes 1, 12 2009.
13. R. LERBOUR, J.-E. MARVIE, AND P. GAUTRON, *Adaptive real-time rendering of planetary terrains*, in Full Paper Proc. Int. Conf. Computer Graphics, Visualization and Computer Vision (WSCG), Feb. 2010, pp. 89–96.
14. MICROSOFT CORPORATION, *Queryperformancecounter function*. [http://msdn.microsoft.com/en-us/library/windows/desktop/ms644904\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms644904(v=vs.85).aspx), 2014.
15. NASA, *NASA World Wind*. <http://goworldwind.org/>, 2004. Accessed 2014-01-15.
16. NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY (NGA), *Department of Defense (DoD) World Geodetic System (WGS) 1984 – Its Definition and Relationships with Local Geodetic Systems*. http://earth-info.nga.mil/GandG/publications/NGA_STND_0036_1_0_0_WGS84/NGA.STND.0036_1.0.0_WGS84.pdf, 7 2014.
17. P. NOWELL, *Mapping a cube to a sphere*. <http://mathproofs.blogspot.de/2005/07/mapping-cube-to-sphere.html>, July 2005.
18. ———, *Mapping a square to a circle*. <http://mathproofs.blogspot.rs/2005/07/mapping-square-to-circle.html>, July 2005.

19. NVIDIA CORPORATION, *EXT texture filter anisotropic*. https://www.opengl.org/registry/specs/EXT/texture_filter_anisotropic.txt, 11 2014.
20. E. O'NEILL AND R. LAUBSCHER, *Extended studies of a quadrilateralized spherical cube earth data base*, Tech. Report NEPRF 3-76 (CSC), Naval Environmental Prediction Research Facility, May 1976.
21. M. RANČIĆ, R. J. PURSER, AND F. MESINGER, *A global shallow-water model using an expanded spherical cube: nomonic versus conformal coordinates*, Q. J. R. Meteorol. Soc, (1996), pp. 959–982.
22. C. RONCHI, R. IACONO, AND P. PAOLUCCI, *The "Cubed Sphere": A New Method for the Solution of Partial Differential Equations in Spherical Geometry*, Journal of Computational Physics, (1996), pp. 96–114.
23. M. SEGAL AND K. AKELEY, *The OpenGL graphics system: A specification, version 4.5 (core profile)*. <https://www.opengl.org/registry/doc/glspec45.core.pdf>, February 2015.
24. J. SNYDER, *Map projections—a working manual*, vol. 1395 of Professional Paper, US Geological Survey, 1987.
25. STACKOVERFLOW, *Mapping a sphere to a cube*. <http://stackoverflow.com/questions/2656899/mapping-a-sphere-to-a-cube>, April 2010.
26. C. C. TANNER, C. J. MIGDAL, AND M. T. JONES, *The clipmap: A virtual mipmap*, in Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98, New York, NY, USA, 1998, ACM, pp. 151–158.
27. T. VINCENTY, *Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations*, Survey Review, 22 (1975), pp. 88–93.
28. L. WILLIAMS, *Pyramidal parametrics*, in Computer Graphics, P. Tanner, ed., vol. 17 of SIGGRAPH '83, ACM, July 1983, pp. 1–11.

Aleksandar M. Dimitrijević
 Faculty of Electronic Engineering, University of Niš
 Department of Computer Science
 Aleksandra Medvedeva 14
 18000 Niš, Serbia
aleksandar.dimitrijevic@elfak.ni.ac.rs

Martin Lambers
 University of Siegen
 Computer Graphics Group
 Hoelderlinstrasse 3
 57076 Siegen, Deutschland
martin.lambers@uni-siegen.de

Dejan D. Rančić
 Faculty of Electronic Engineering, University of Niš
 Department of Computer Science

Aleksandra Medvedeva 14
18000 Niš, Serbia
dejan.rancic@elfak.ni.ac.rs