

REFEREED PAPER

Unfolding the Earth: Myriahedral Projections

Jarke J. van Wijk

Dept. of Mathematics and Computer Science, Technische Universiteit Eindhoven, Eindhoven, The Netherlands
Email: vanwijk@win.tue.nl

Myriahedral projections are a new class of methods for mapping the earth. The globe is projected on a myriahedron, a polyhedron with a very large number of faces. Next, this polyhedron is cut open and unfolded. The resulting maps have a large number of interrupts, but are (almost) conformal and conserve areas. A general approach is presented to decide where to cut the globe, followed by three different types of solution. These follow from the use of meshes based on the standard graticule, the use of recursively subdivided polyhedra and meshes derived from the geography of the earth. A number of examples are presented, including maps for tutorial purposes, optimal foldouts of Platonic solids, and a map of the coastline of the earth.

INTRODUCTION

Mapping the earth is an old and intensively studied problem. For about two thousand years, the challenge to show the round earth on a flat surface has attracted many cartographers, mathematicians, and inventors, and hundreds of solutions have been developed. There are several reasons for this high interest. First of all, the geography of the earth itself is interesting for all its inhabitants. Secondly, there are no perfect solutions possible such that the surface of the earth is depicted without distortion. Finally, factors such as the intended use of the map (e.g. navigation, visualisation, or presentation), the available technology (pen and ruler or computer), and the area or aspect to be depicted lead to different requirements and hence to different optima.

A layman might wonder why map projection is a problem at all. A map of a small area, such as a district or city, is almost free of distortion. So, to obtain a map of the earth without distortion one just has to stitch together a large number of such small maps. In this article we explore what happens when this naive approach is pursued. We have coined the term *myriahedral* for the resulting class of projections. A *myriahedron* is a polyhedron with a myriad of faces. The Latin word *myriad* is derived from the Greek word *myrioi*, which means ten thousands or innumerable. We project the surface of the earth on such a myriahedron, we label its edges as folds or cuts, and fold it out to obtain a flat map.

In the next section, some basic notions on map projection are presented, and related work is shortly described, followed by a section in which an overview of the approach employed here is given. Different solutions

are obtained by using different myriahedra and choices for the edges to be cut, which are described in three separate sections. The use of graticule-based meshes, recursively subdivided polyhedra, and geographically aligned meshes lead to different maps, each with their own strengths. Finally, the results are discussed.

BACKGROUND

The globe is a useful model for the surface of the earth. Locations on a globe (and the earth) are given by latitude ϕ and longitude λ . The position of a point $p(\phi, \lambda)$ on a globe with unit radius is $(\cos\lambda\cos\phi, \sin\lambda\cos\phi, \sin\phi)$. Curves of constant ϕ , such as the equator, are parallels; curves of constant λ are meridians. A graticule is a set of parallels and meridians at equal spacing in degrees.

Compared with a map, a globe has some disadvantages, such as poor portability, and to obtain a more practical solution, the spherical globe has to be mapped to a flat surface. This puzzle has intrigued many researchers for two thousand years. John P. Snyder has provided a fascinating overview of the history of map projection. In the following, references for map projections are only given if not discussed in his book (Snyder, 1993), to keep the number of references within bounds. Introductions to map projection can be found in textbooks on cartography or geographic visualisation (Robinson *et al.*, 1995; Kraak and Ormeling, 2003; Slocum *et al.*, 2003). Also on the web much information can be found, for instance in the extensive website developed by Furuti (2006).

The major problem of map projection is distortion. Consider a small circle on the globe. After projection on a

map, this circle transforms into an ellipse, known as the Tissot indicatrix, with semi-axes with lengths a and b . If $a = b$ for all locations, then angles between lines on the globe are maintained after projection: The projection is *conformal*. The classic example is the Mercator projection. Locally, conformality preserves shapes, but for larger areas distortions occur. For example, in the Mercator projection shapes near the poles are strongly distorted.

If $ab = C$ for all locations on the map, then the projection has the *equal-area* property: Areas are preserved after projection. Examples are the sinusoidal, Lambert's cylindrical equal area and the Gall-Peters projection.

The problem is that for a double curved surface no projection is possible that is both conformal and equal-area. Along a curve on the surface, such as the equator, both conditions can be met; however, at increasing distance from such a curve the distortion accumulates. Therefore, depending on the purpose of the map, one of these properties or a compromise between them has to be chosen. Concerning distortion, uniform distances are another aspect to be optimised. Unfortunately, no map projections are possible such that distances between any two positions are depicted on a similar scale, but one can aim at small variations overall or at proper depiction along certain lines.

Besides these constraints from differential geometry, map projection also has to cope with a topological issue. A sphere is a surface without a boundary, whereas a finite flat area has to be bounded. Hence, a cartographer has to decide where to cut the globe and to which curve this cut has to be mapped. Many choices are possible. One option, used for azimuthal projections, is to cut the surface of the globe at a single point, and to project this to a circle, leading to very strong distortions at the boundary. The most popular choice is to cut the globular surface along a meridian, and to project the two edges of this cut to an ellipse, a flattened ellipse or a rectangle, where in the last two cases the point-shaped poles are projected to curves.

The use of interrupts reduces distortion. For the production of globes, minimal distortion is vital for production purposes; hence gore maps are used, where the world is divided in for instance twelve gores. Goode's homolosine projection (1923) is an equal-area projection, composed from twelve regions to form six interrupted lobes, with interrupts through the oceans. The projection of the earth on unfolded polyhedra instead of rectangles or ellipses is an old idea, going back to Da Vinci and Dürer. All regular polyhedra have been proposed as suitable candidates. Some examples are Cahill's Butterfly Map (1909, octahedron) and the Dymaxion Map of Buckminster Fuller, who used a cuboctahedron (1946) and an icosahedron (1954). Steve Waterman has developed an appealing polyhedral map, based on sphere packing.

Figure 1 visualises the trade-off to be made when dealing with distortion in map projection. An ideal projection should be equal-area, conformal, and have no interrupts; however, at most, two of these can be satisfied simultaneously. Such projections are shown here at the corners of a triangle, whereas edges denote solutions where one of the requirements is satisfied. Existing solutions can be positioned in this solution space. Examples are given for some

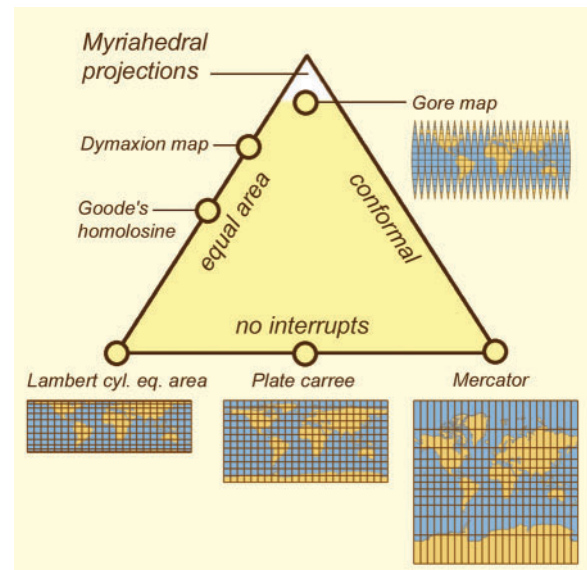


Figure 1. Distortion in map projection

cylindrical projections, with linear parallels and meridians. Most of the existing solutions, using no interrupts, are located at the bottom of the triangle. In this article, we explore the top of the triangle, which is still *terra incognita*, using geographic terminology. Or, in other words, we discuss projections that are both (almost) equal area and conformal, but do have a very large number of interrupts.

Related issues have been studied intensively in the fields of computer graphics and geometric modelling, for applications such as texture mapping, finite-element surface meshing, and generation of clothing patterns. The problem of earth mapping is a particular case of the general surface parameterisation problem. A survey is given by Floater and Hormann (2005). Finding strips on meshes has been studied in the context of mesh compression and mesh rendering, for instance by Karni *et al.* (2002). Bounded-distortion flattening of curved surfaces via cuts was studied by Sorkine *et al.* (2002). The work presented here has a different scope and ambition as this related work. The geometry to be handled is just a sphere. The aim is to obtain zero distortion, and we accept a large number of cuts. Finally, we aim at providing an integrated framework, offering fine control over the results, and explore the effect of different choices for the depiction of the surface of the earth.

METHOD

We project the globe on a polyhedral mesh, label edges as cuts or folds, and unfold the mesh. We assume that the faces of the mesh are small compared with the radius of the globe, such that area and angular distortion are almost negligible. We first discuss the labelling problem. A mesh can be considered as a (planar) graph $G = (V, E)$, consisting of a set of vertices V and undirected edges E that connect vertices. Consider the dual graph $H = (V', E')$, where each vertex denotes a face of the mesh, and each edge corresponds to an edge of the original graph, but now

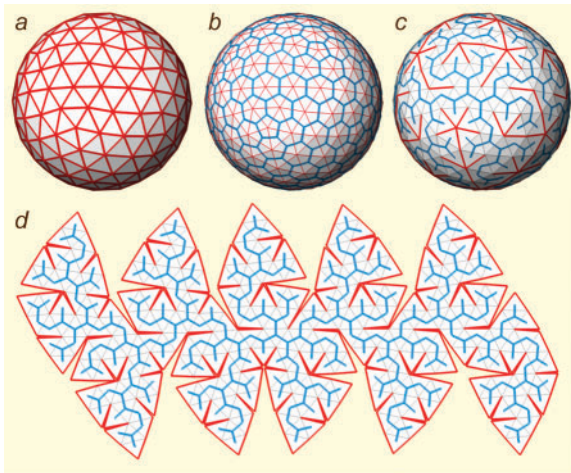


Figure 2. (a) Mesh G ; (b) Dual mesh H ; (c) Cuts and folds; (d) Foldout

connecting two faces instead of two vertices (Figure 2). After labelling edges as folds and cuts, we obtain two subgraphs H_f and H_c , where all edges of each subgraph are labelled the same. The labelling of edges should be done such that

- the foldout is connected. In other words, in H_f a path should exist from any node (face of the mesh) to any other node.
- the foldout can be flattened. Hence, in H_f no cycles should occur, otherwise this condition cannot be met.

Taken together, these constraints imply that H_f should be a spanning tree of H . Also, the subgraph G_c of G with only edges labelled as cuts should be a spanning tree of G . This can be seen as follows. All vertices should have one or more cuts in the set of neighbouring edges (otherwise the foldout can not be flattened), and cycles in the cuts would lead to a split of the foldout. The set of cuts unfolds to a single boundary, with a length of twice the sum of lengths of the cuts.

There is a third constraint to be satisfied: The labelling should be such that the foldout does not suffer from fold-overs. The folded out mesh should not only be planar, it should also be single-valued. The use of an arbitrary spanning tree does lead to fold-overs in general. However, we found empirically that the schemes we use in the following almost never lead to fold-overs, and we do not explicitly test on this. The problem of fold-overs is complex, and we cannot give proofs on this. Nevertheless, it can be understood that fold-overs are rare by observing that the sphere is a very simple, uniform, convex surface; and also, the typical patterns that emerge are strips of triangles, connected to and radiating outward from a line or point, which strips rarely overlap.

The term spanning tree suggests a solution for labelling the edges: Minimal spanning trees of graphs are a well-known concept in computer science. Assign a weight $w(e_i)$ to each of the edges e_i , such that a high value indicates a high strength and that we prefer this edge to be a fold. Next, calculate a maximal spanning tree H_f (or a minimal spanning tree G_c), *i.e.*, a spanning tree such that the sum of

the weights its edges is maximal (or minimal). The algorithm to produce a myriahedral projection is now as follows:

1. Generate a mesh;
2. Assign weights to all edges;
3. Calculate a maximal spanning tree H_f ;
4. Unfold the mesh;
5. Render the unfolded mesh.

In the following sections, we discuss various choices for the first two steps, here we describe the last three steps, which are the same for all results shown.

For the calculation of the maximal spanning tree we followed the recommendations given by Moret and Shapiro (1991). We use Prim's algorithm (Prim, 1957) to find a maximal spanning tree. Starting from a single vertex, iteratively, the neighbouring edge with the highest weight and the corresponding vertex is added. This gives an optimal solution. The neighbouring edges of the growing tree are stored in a priority queue, for which we use pairing heaps (Fredman, Sedgwick, Sleator and Tarjan, 1986). The performance is $O(|E| + |V| \log |V|)$, where $|E|$ and $|V|$ denote the number of edges and vertices. In practice, optimal spanning trees are calculated within a second for graphs with ten thousands of edges and vertices.

Unfolding is straightforward. Assume that all faces of the mesh are triangles. Faces with more edges can be handled by inserting interior edges with very high weights, such that these faces are never split up. Unfolding is done by first picking a central face, followed by recursive processing of adjacent faces. Consider two neighbouring triangles PQR and RQS , and assume that the unfolded positions P' , Q' , and R' are known. Next, the angle α between RQS and the plane of PQR is determined, and S' is calculated such that the new angle is α' , $|QS| = |Q'S'|$ and $|RS| = |R'S'|$. The use of $\alpha' = 0$ gives a flat mesh, use of (for instance) $\alpha' = \alpha(1 + \cos(\pi t/T))/2$ gives a pleasant animation (examples are shown in <http://www.win.tue.nl/~vanwijk/myriahedral>).

The geography of the earth (or whatever image on a spherical surface has to be displayed) is mapped as a texture on the triangles. We use the maps of David Pape for this (Pape, 2001). When the triangles are large compared with the radius of the globe, like in standard polyhedral projections, the triangles have to be subdivided further to control the projection in the interior. We use a simple gnomonic projection here.

Rendering maps for presentation purposes requires proper anti-aliasing, because regular patterns and very thin gaps have to be dealt with. For the images shown, 100-fold supersampling per pixel with a jittered grid was used, followed by filtering with a Mitchell filter.

All images were produced with a custom developed, integrated tool to define meshes and weights, and to calculate and render the results, running under MS Windows. Response times on standard PCs range from instantaneous to a few seconds, which enables fast exploration of parameter spaces. Rendering of high resolution, high quality maps can take somewhat longer, up to a few minutes.

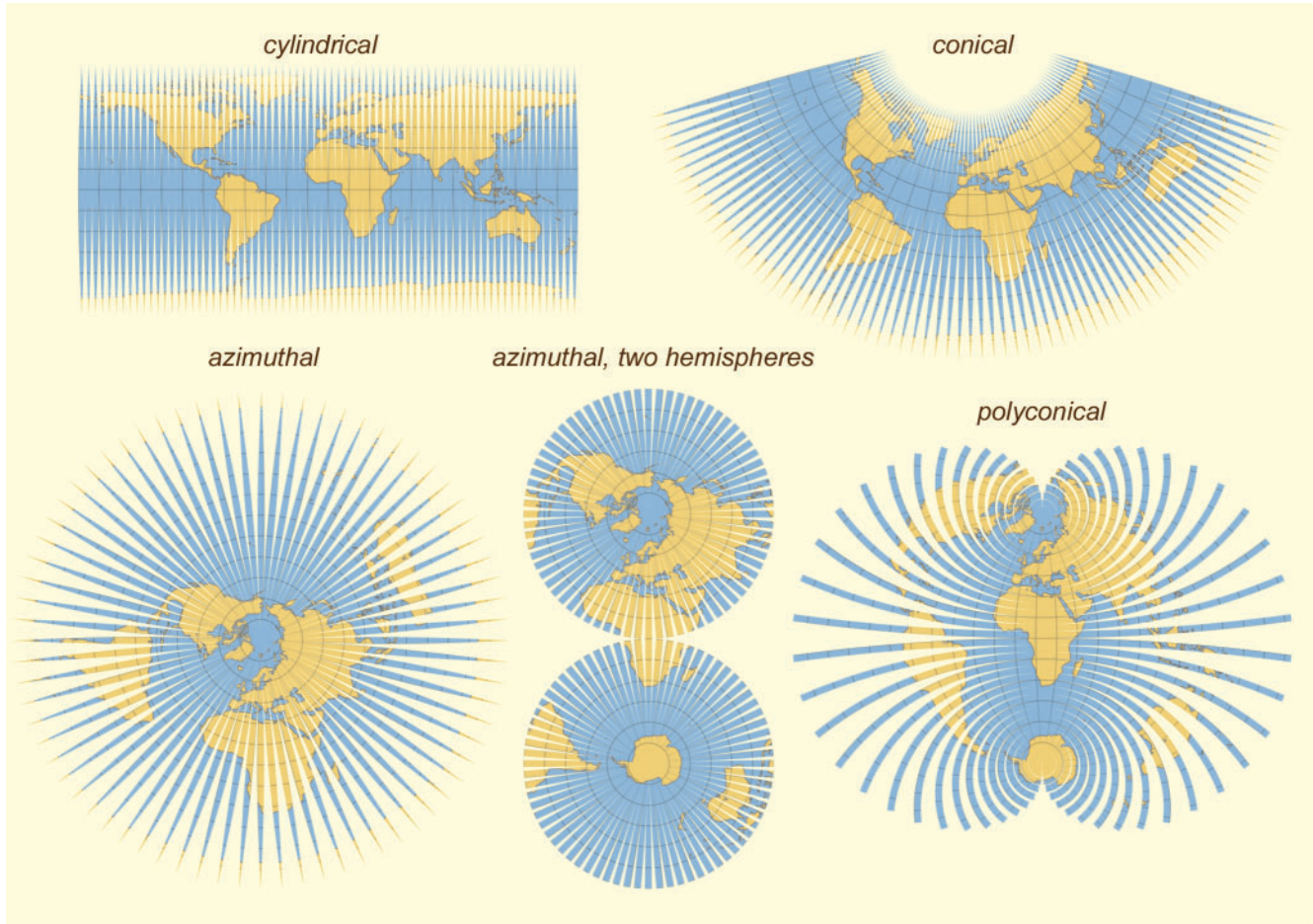


Figure 3. Graticular projections, derived from a 5° graticule. 2592 polygons: a) cylindrical; b) conical; c) azimuthal; d) azimuthal, two hemispheres; e) polyconical

GRATICULES

The simplest way to define a mesh is to use the graticule itself, and to cut along parallels or meridians. The results can be used as an introduction to map projection. A weight for edges, using the value of ϕ and λ of the midpoint of an edge, can be defined as

$$w(\phi, \lambda) = -(W_\phi |\phi - \phi_0| + W_\lambda \min_k |\lambda - \lambda_0 + 2\pi k|),$$

where W_ϕ and W_λ are overall scaling factors, and ϕ_0 and λ_0 denote where a maximal strength is desired. Different values for these lead to a number of familiar looking projections (Figure 3). The use of a high value for W_ϕ gives cuts along meridians. Dependent on the value of ϕ_0 a cylindrical projection (0° , equator), an azimuthal projection (90° , North pole), or a conical projection (here 25°) is obtained when the meridian strips are unfolded. Use of a negative value for W_ϕ gives two hemispheres, each with an azimuthal projection. The meridian at which to be centred can be controlled by using a low value for W_λ and a suitable value for λ_0 . The use of a high value for W_λ gives cuts along parallels. Unfolding these parallels gives a result resembling the polyconic projection of Hassler (1820).

The relation between a spatially varying weight w and the decision where to cut and fold can be understood by considering Prim's algorithm. Suppose, without loss of generality, that we start at a maximum of w and proceed to attach the edges with the highest weight. At some point, edges at the boundary will have approximately the same weight and, after a number of additions, a ring of faces is added, with cuts in between neighbouring faces in this ring. Hence, edges aligned with contours of w typically turn into folds, whereas edges aligned with gradients of w turn into cuts.

Each strip is almost free of angular or area distortion, however, a large number of interrupts occur with varying widths. These gaps visualise, just like the Tissot indicatrix, the distortion that occurs when a non-interrupted map is used, and can be used to explain the basic problem of map projection. If we want to close these gaps, the strips must be broadened. However, to maintain an equal area, they have to be shortened, and to maintain the same aspect ratio they have to be lengthened, which is not possible simultaneously. Also, it is clearly visible that mapping a point (such as a pole) to a line leads to a strong distortion.

When the number of strips is increased, the gaps are less visible, and the distortion is shown via the transparency of the map (Figure 4).

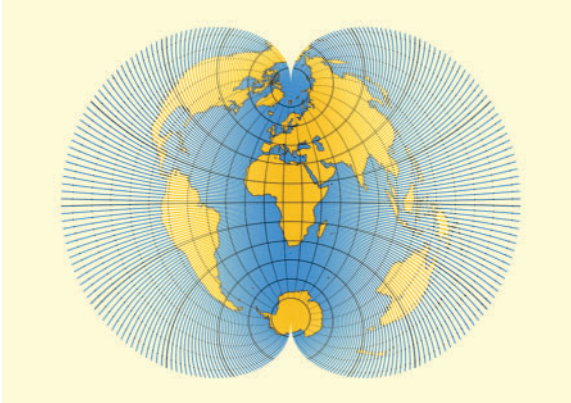


Figure 4. Polyconical projection, derived from a 1° graticule, 64 800 polygons

RECURSIVE SUBDIVISION

For the graticular projections, thin strips of faces are attached to one single strip or face. This is a degenerated tree structure. In this section, we consider what results are obtained when a more balanced pattern is used. To this end, we start with Platonic solids for the projection of the globe, and recursively subdivide the polygons of these solids. This approach has been used before for encoding and handling geospatial data (Dutton, 1996).

At each level i , each edge is split and the new centres, halfway on the greater circle connecting the original endpoints, are connected. As a result, for instance each triangle is replaced at each level by four smaller triangles. Other subdivision schemes can also be used, for instance triangles can be subdivided into nine smaller ones.

The edge weights are set as follows. We associate with each edge three numbers w_0 , w_1 , and w_c , where the first two correspond with the endpoints and the latter with the centre position. For new edges, $w_0 \leftarrow i$, $w_1 \leftarrow i$, and $w_c \leftarrow i+1$. If an edge e is split into two edges e' and e'' , we use linear interpolation for the new values

$$w'_0 \leftarrow w_0, \quad w'_1 \leftarrow w_c, \quad w'_c \leftarrow (w_0 + w_c)/2;$$

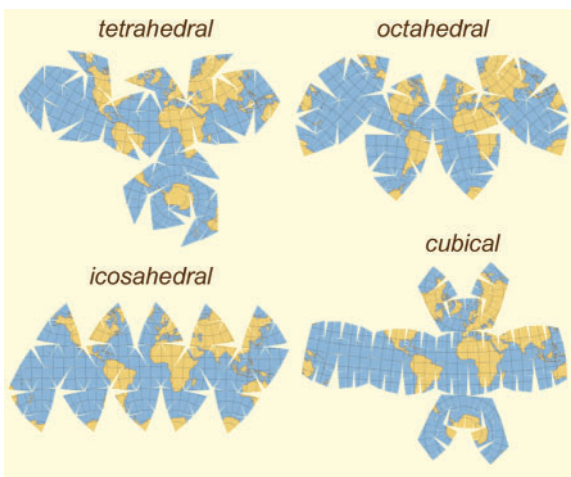


Figure 5. Recursive subdivision of Platonic solids, using five levels of subdivision, 4096–20 480 polygons

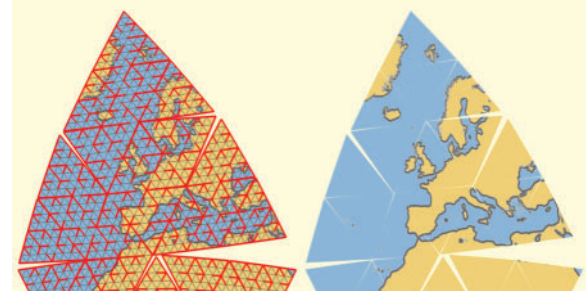


Figure 6. Close-up of icosahedral projection

$$w''_0 \leftarrow w_c, \quad w''_1 \leftarrow w_1, \quad w''_c \leftarrow (w_c + w_1)/2.$$

As a result, the weights are highest close to the centre of original edges. Finally, we use w_c as the edge weight for the edges of the final mesh, plus a graticule weight w with small values for W_λ and W_ϕ to select the aspect.

The resulting unfolded maps are, at first sight, somewhat surprising (Figure 5). One would expect to see interesting fractal shapes, however, at the second level of subdivision the gaps are already almost invisible (Figure 6). Indeed, the structure of the cuts is self-similar, however, for higher levels of subdivision and smaller triangles, the surface of the sphere quickly approaches a plane, which has Hausdorff dimension 2. Only when areas would be removed, such as the centre triangles in the Sierpinski triangle, a fractal shape would be obtained.

As a step aside, fractal surfaces and foldouts do not match well either. Unfolding, for instance, a recursively subdivided surface with displaced midpoints leads to a large number of fold-overs (Figure 7).

As another step aside, let us consider optimal mapping on Platonic solids. We consider a map optimal when the cuts do not cross continents. To find such mappings, we assign to each edge a weight proportional to the amount of land cut, computed by sampling the edges at a number of positions (here we used 25) and looking up if land or sea is covered in a texture map of the earth. Next, the map is unfolded using the standard method and the sum of weights of cut edges is determined. This procedure is repeated for a large number of orientations of the mesh, searching for a minimal value. We used a sequence of three rotations to vary the orientation of the mesh, and used steps of 1° per rotation. Results are shown in Figure 8.

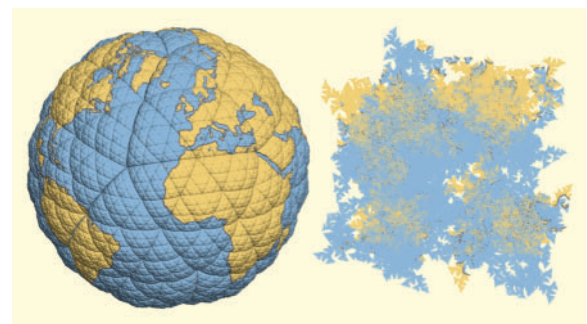


Figure 7. Folding out a fractal surface gives a mess

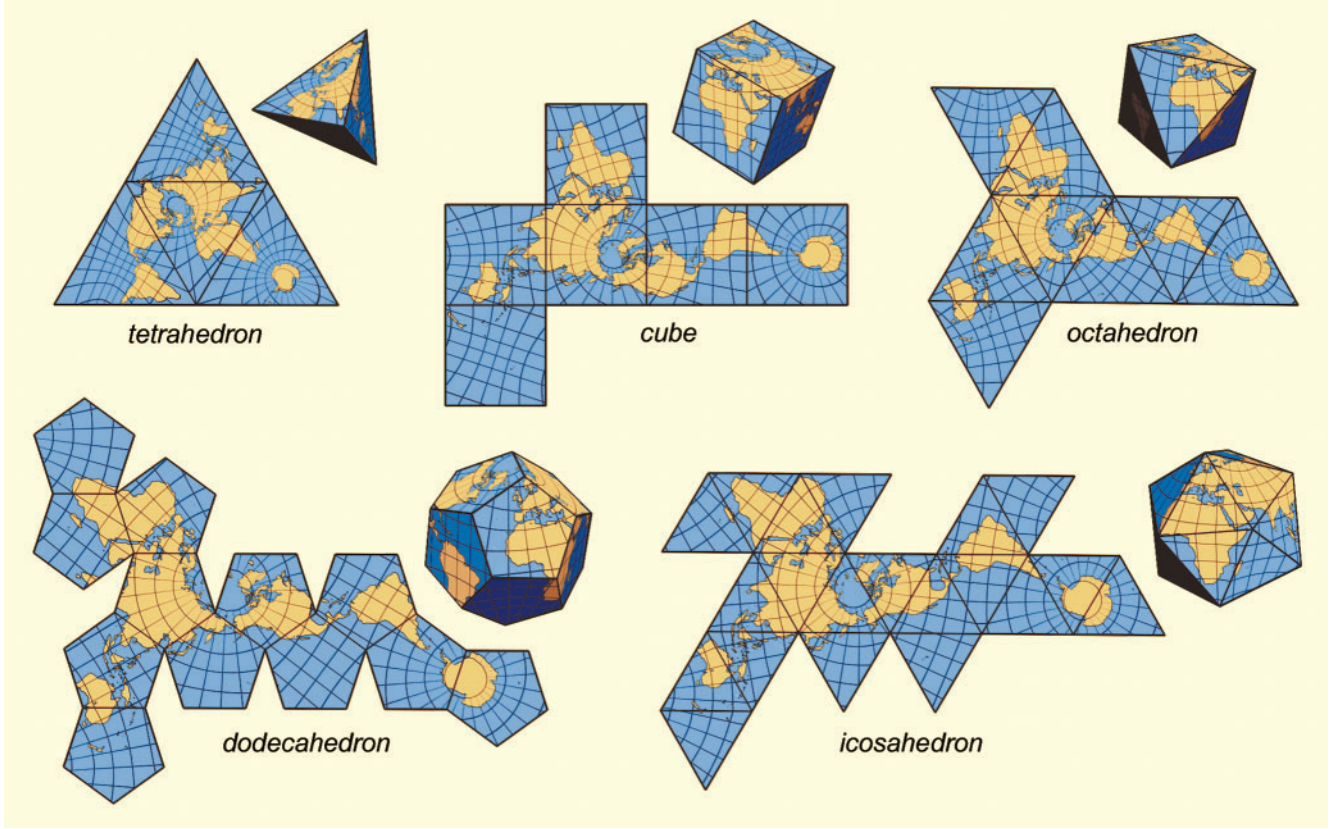


Figure 8. Optimal fold-outs of Platonic solids

For the tetrahedron a perfect, and for the other platonic solids an almost perfect, mapping is achieved. Except for the tetrahedron, the resulting layout of the continents is the same as the layout used by Fuller for his Dymaxion map. He used a slightly modified icosahedron for his best-known version, but the version shown here reveals that his modifications are not necessary per se.

GEOGRAPHY ALIGNED MESHES

Taking continents into account when deciding where to cut is an obvious idea. In this section, we explore this further. We generate meshes such that continents are cut orthogonal to their boundaries. First, we define for each point on the sphere a value $f(\phi, \lambda)$ that denotes the amount of land in its neighbourhood. High values are in the centres of continents, low values in the centres of oceans. This function is used to generate the mesh, and also to control the strength of edges. We use linear interpolation of a matrix of values F_{ij} , with $i = 0, \dots, I-1$ and $j = 0, \dots, J-1$ to calculate $f(\phi, \lambda)$. The corresponding values for λ and ϕ per element are $\lambda_i = 2\pi(i+0.5)/I$ and $\phi_j = \pi(j+0.5)/J - \pi/2$, respectively. The matrix F is derived from a raster image R of a map with the same dimensions as F via convolution with a filter m , i.e.,

$$F_{ij} = \sum_{k=K_j^-}^{K_j^+} \sum_{l=L^-}^{L^+} m_{jkl} R_{i+k, j+l},$$

where $i+k$ is calculated modulo I , $L^- = \max(-j, -L)$, and $L^+ = \min(J-1-j, L)$. We typically use $I = 256$, $J = 128$, and $L = 32$. A large weight mask m is used, because it is not only the edges that have to be blurred, but also areas far from coastlines must be assigned varying values. The convolution has to be done taking the curvature into account; therefore, the width and contents of the mask have to be adapted per scan line. For the width, we use $K_j^+ = -K_j^- = \lceil IL/2J \cos \phi_j \rceil$. We use a Gaussian filter, taking the distance r_{jkl} along a greater circle into account between a centre element $R_{0,j}$ and an element $R_{k,j+l}$, as well as the area a_{jl} of the latter. Specifically,

$$m_{jkl} = \sum_{k=K_j^-}^{K_j^+} \sum_{l=L^-}^{L^+} s_{jkl},$$

with

$$s_{jkl} = a_{jkl} \exp(-r_{jkl}^2/2\sigma^2)/\sqrt{2\pi}\sigma,$$

$$a_{jl} = 2\pi^2 \cos \phi_{j+l}/NM, \text{ and}$$

$$r_{jkl} = \arccos \left[\mathbf{p}(\phi_j, 0) \cdot \mathbf{p}(\phi_{j+l}, \lambda_k) \right].$$

Figure 9 shows an example. As a result, for instance the value for the South Pole is similar to that of the centre of South America.

To obtain a foldout with cuts perpendicular to contours of f , the following steps are performed (Figure 10), inspired

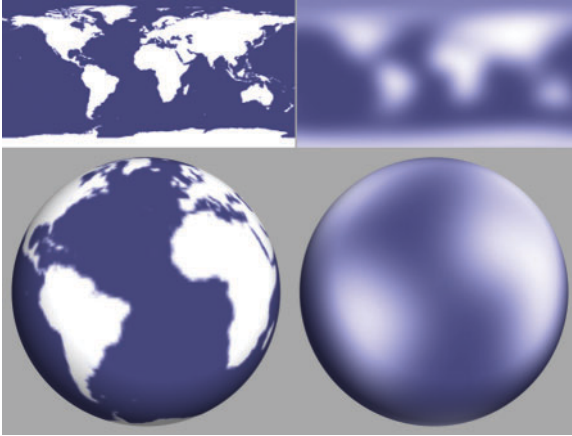


Figure 9. From R to F via convolution with a Gaussian

by the anisotropic polygonal remeshing method of Alliez *et al.* (2003):

- Generate mesh lines along and perpendicular to contours of f with the algorithm of Jobard and Lefer (Jobard and Lefer, 1997);
- Calculate intersections of these sets of lines, and derive polygons;
- Tessellate polygons with more than four edges; and finally
- Use the standard approach to decide on folds and cuts.

These steps are discussed in more detail.

The algorithm of Jobard and Lefer (Jobard and Lefer, 1997) is an elegant and fast method to produce equally spaced streamlines for a given vector field. Starting from a single streamline, new streamlines are repeatedly started from seedpoints at a distance d from points of existing streamlines, and traced in both directions. If such a streamline is too close to an existing streamline or when a cycle is formed, the tracing is stopped. The time critical step is to determine which points are close. The standard solution is to use a rectangular grid for fast look-up. Here streamlines are traced in (ϕ, λ) space, and the mapping to the sphere has to be taken into account. We therefore use horizontal strips of rectangles, where the number of rectangles per strip is proportional to $\cos\phi$.

To obtain mesh lines along contours, the vector field

$$c(\phi, \lambda) = (f_\lambda, -f_\phi) / \sqrt{\cos^2 \phi f_\phi^2 + f_\lambda^2}$$

is traced; lines perpendicular to contours follow from tracing the vector field

$$g(\phi, \lambda) = (f_\phi \cos \phi, f_\lambda / \cos \phi) / \sqrt{\cos^2 \phi f_\phi^2 + f_\lambda^2}, \text{ where}$$

$$f_\lambda = \partial f(\phi, \lambda) / \partial \lambda \text{ and } f_\phi = \partial f(\phi, \lambda) / \partial \phi.$$

The factors $\cos\phi$ in the definition of c and g follow from the requirements that we want these fields to have a unit magnitude and to be orthogonal after projection on the sphere. Projection implies that components $\Delta\lambda$ of a vector $(\Delta\phi, \Delta\lambda)$ are scaled with a factor $\cos\phi$, whereas the $\Delta\phi$ components keep their length. For the tracing, we use a fourth order Runge–Kutta method with a fixed time step.

In the next step, crossings between these line sets are calculated and the lines are cleaned up. Streamlines without crossings are removed, neighbouring points of crossings are removed from the streamlines, and heads and tails are removed. Next, the resulting net is scanned and a set of polygons, covering the sphere, is constructed. This gives a regular, rectangular mesh for a large part of the sphere, but also and unfortunately, irregular polygons. This can be understood from the topology of vector fields, a well-known topic in the visualisation community (Helman and Hesselink, 1991). Critical points are points where the magnitude of the vector field is zero. For the vector fields used here, these occur at maxima of f (centres of continents), minima of f (centres of oceans) and at saddle-points of f (for instance between South America and Africa). The domain of a flow field can be tessellated using streamlines between these critical points, the so-called separatrices, which gives a topological decomposition of the domain. For the vector fields used here, separatrices typically run through valleys of f . When f is used to decide which edges to label as cuts, the surface breaks along those valleys, which in turn appear as overall boundaries. Downhill gradient lines of f , following g , bend into such valleys with a sharp turn or stop because a line at the other side is too close, leading to irregular polygons.

We use a standard triangulation algorithm to tessellate polygons with more than four edges. First, the polygon is split into convex polygons, next, triangles are split off. Heuristics used are a preference for short inserted edges and avoidance of obtuse or very sharp angles. This is not perfect yet and leads to a somewhat fractured and irregular appearance of the map when unfolded. Improvement turns out not to be simple. In an image like Figure 10(c) it is easy

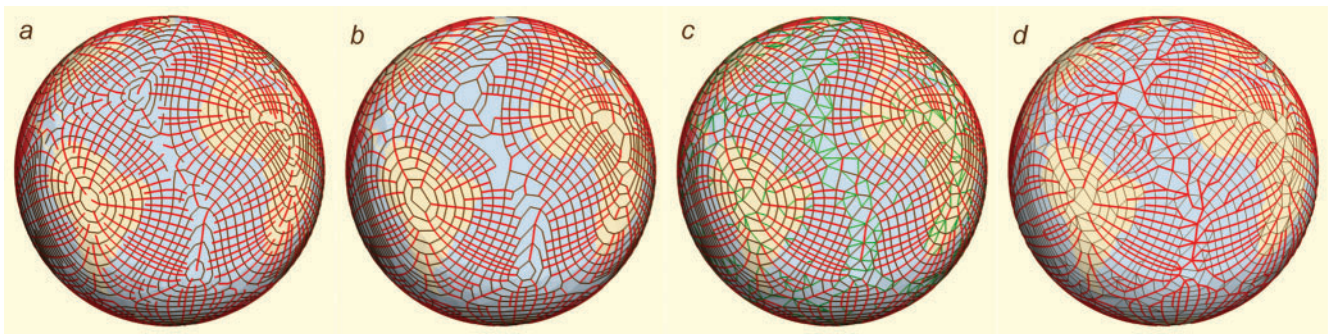


Figure 10. Use of contours and gradients to derive a mesh: a) Jobard and Lefer algorithm; b) finding polygons; c) triangulation; d) deciding on cuts

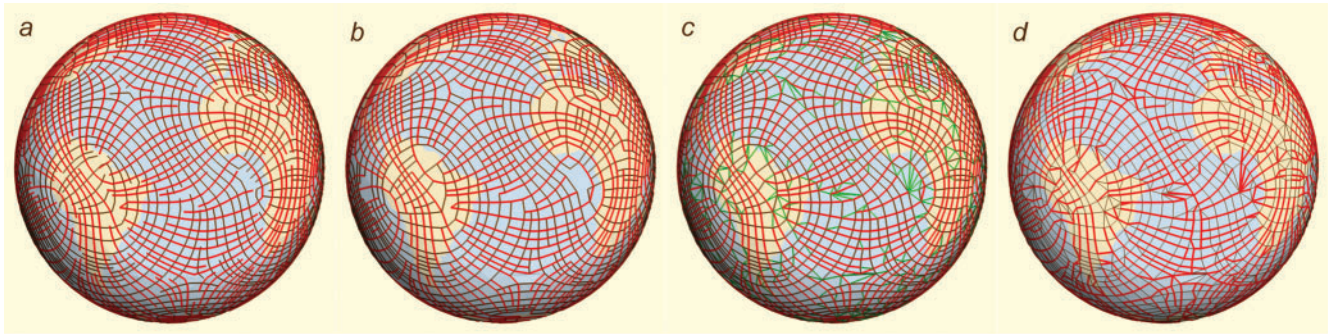


Figure 11. Same as Figure 10, using curvature tensor field

to point at polygons where better choices could have been made, the hard part is to find methods that have no adverse effect at other locations. For instance, introducing extra points and edges often leads to more irregularities, and tracing lines between critical points in advance gives wide interrupts instead of multiple smaller ones.

We also tried the use of a tensor field based on curvature (Alliez *et al.*, 2003), instead of a vector field (Figure 11). Here, at each point, the direction of minimum or maximum curvature is traced. This gives an orthogonal mesh, without singularities along lines and, indeed, the valley in the centre of the Atlantic Ocean is now filled in a more regular way. However, this does not necessarily lead to a more appealing tessellation, see for instance the small strip introduced in the centre of this valley. Tensor fields have two kinds of singular points: trisectors and wedges. Here, a trisector appears in the northern Atlantic Ocean, and a wedge in the Gulf of Guinea. This latter feature leads to irregularities in the resulting mesh.

Other solutions are to increase the density of the mesh, and, simply to accept the fractured boundaries. Visually, they show that the surface of the globe is torn apart, and they show that where this is done exactly is somewhat arbitrary.

Figure 12 shows results of this approach. Straightforward application leads again to the layout of the continents of Buckminster Fuller. A more familiar layout can be obtained by adding a graticular weight, and tuning W_λ and W_ϕ . The overall layout resembles a conical projection. The continents are shown with few interrupts and with correct shape and relative position. Instead of f , also $|f - f_c|$ can be used as a weight for the edges. As a result, the global boundary of the map is along contours $f = f_c$. This boundary is smooth, and divides the surface here into the main continents, the oceans, and Antarctica. The author does not know a similar map.

Also, $-f$ can be used as a weight for the edges. This results in a map where the oceans are central, surrounded by the coastline of the world. Ocean centred maps have been made before, such versions are available for Goode's homolosine map and Fuller's Dymaxion map. Closest is a map presented by Athelstan Spilhaus (Spilhaus, 1983). His map (and also Fuller's) is centred on Antarctica, showing the oceans as three lobes, and is, hence, somewhat less extreme than the version shown here. A map similar to Spilhaus's map can easily be generated with our method, simply by removing Antarctica from the map R .

DISCUSSION

We have presented a new class of map projections, based on projecting the earth on myriahedra, polyhedra with many faces, and unfolding these. A general approach is presented to decide on cuts and folds, based on weighting the edges and calculating a maximal spanning tree. Three different choices for types of meshes and weighting schemes are presented, leading to a variety of different projections of the surface of the earth.

There remains one question to be answered: What is this all good for? Most resulting maps are highly unusual, and do not correspond with what on average is considered to be a useful map.

Furthermore, the complexity is high. Standard projection methods require, in the worst case, a few iterations per point to solve a transcendental equation; the methods presented here require implementation of a number of non-trivial algorithms. Hence, forward mapping is not easy, and also inverse mapping, from a location on the map to a point on the globe, is much more involved than with standard maps.

Fortunately, there are also positive aspects that can be mentioned. From an academic point of view, a classic topic like map projection deserves an exhaustive exploration and this class of maps has not been addressed yet. What happens when many small maps are glued together is obvious and here an extensive answer is given. Hence, these maps could be used for textbook purposes. Furthermore, each class has its own interesting aspects. The graticular maps can be used to explain the basics of map projection. Polyhedral maps are entertaining, and here we have presented optimal versions.

We have investigated what happens when interrupts are removed. In Figure 13, two examples are shown, derived from maps shown in Figure 12. We matched corresponding vertices at a distance below a certain threshold, starting at the ends of gaps, followed by a finite element simulation to redistribute the points of the mesh. In the examples shown, we defined the stiffness matrix such that the equal-area property is satisfied. These steps are repeated until no corresponding vertices could be found. The maps are not conformal: Parallels and meridians do not cross at right angles. The hard boundaries of the maps without interrupts are somewhat arbitrary, but do attract attention, in contrast to the more fuzzy boundaries of their myriahedral counterparts. Finally, they reveal a quality of all myriahedral maps. The interrupts present in myriahedral maps show the inevitable distortion in a natural, and explicit way, whereas

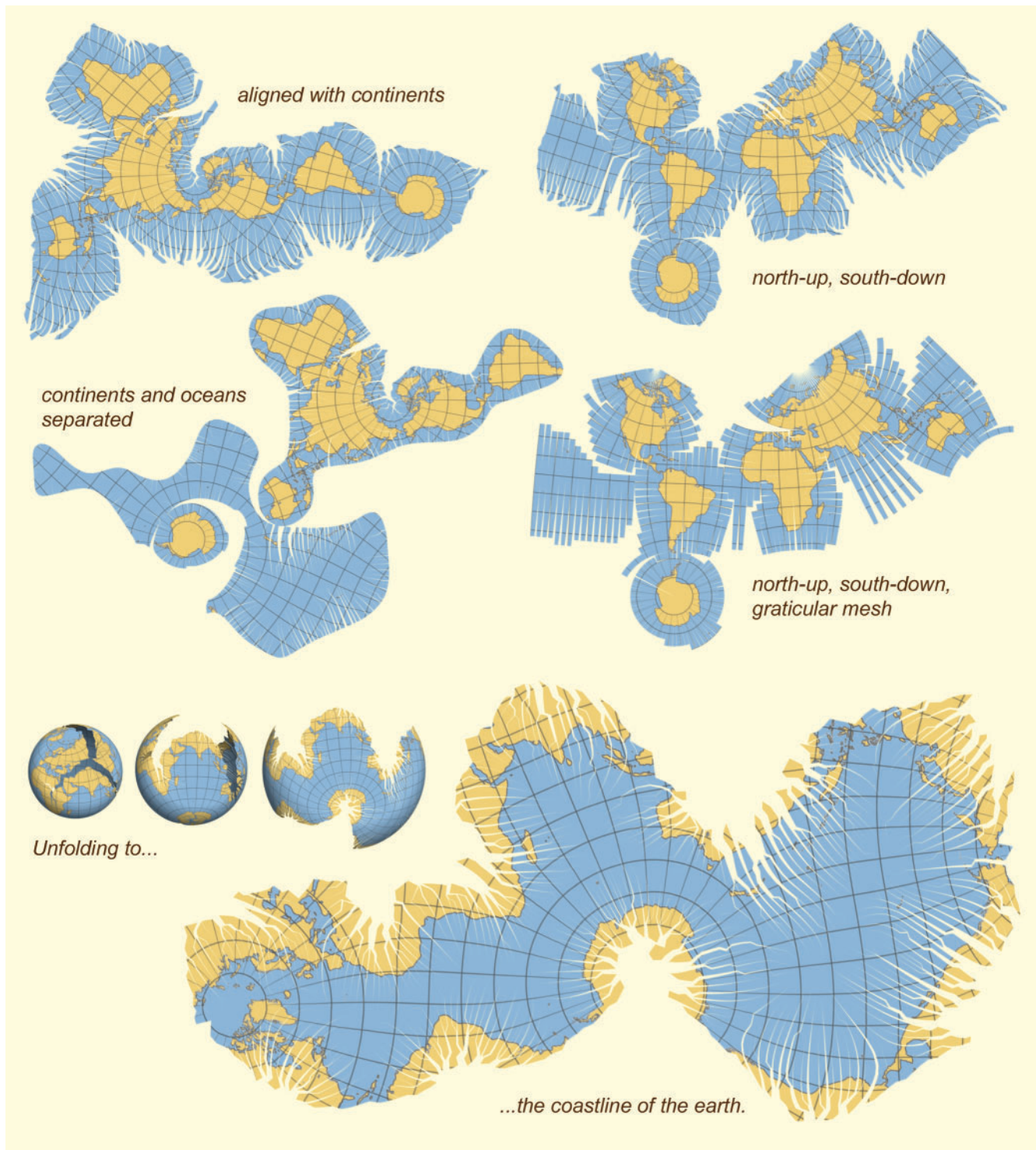


Figure 12. Myriahedral projections with geography aligned meshes, 5500 polygons

in standard maps it is left to the viewer to guess where and which distortion occurs.

Methodologically interesting is that here a computer science approach is used, whereas map projection is traditionally the domain of mathematicians, cartographers, and mathematical cartographers. Myriahedral projections are generated using algorithms, partially originating from flow visualisation, and not by formulas. Implementation is

not simple, but when the machinery is set up, a very large variety of maps can be generated just by changing parameters, such as W_λ , W_ϕ , F , f_0 , σ , and the size of the faces used. This leaves much room for serendipity, and indeed, some of the maps shown here were discovered by accident.

Maps are not only used for navigation or visualisation, but also for decorative, illustrative and even rhetoric

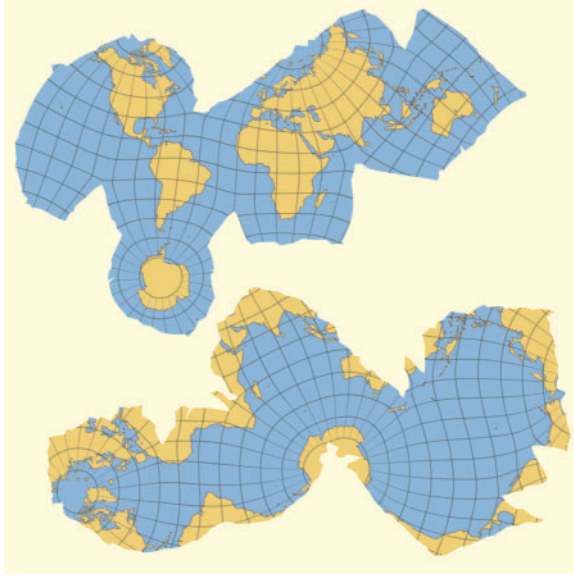


Figure 13. Closed gaps

purposes (Monmonier, 1991), for instance on covers of magazines. The coastline map is an example, which can serve to emphasise the importance of oceans. Another example is shown in Figure 14. Here, we used a subdivided icosahedral projection, centred at 40°N , 100°W , and used an edge weight proportional to the distance from this point, plus a small random factor.

As a first step to a user study, the resulting projections were shown as static images and in animated version to about 20 people, ranging from laymen via computer scientists to cartographers. In general, the reception was very positive. Most people found the results compelling and intriguing. Computer science colleagues liked the general framework and the algorithms. Nevertheless, taking a utilitarian point of view, some cartographers argued that cuts are always more disturbing reading a map than having distortion, which is the reason that such projections have been discarded so far and are not useful in practice. Concerning usability for tutorial purposes, results were mixed again. Some cartographers found this a very strong feature; others argued that visualising a distorted grid would be more effective. More elaborate usability tests are required to evaluate which approach is most effective here and, also, to see what the value is in general. Besides such a lab test, an interesting test is whether these results are interesting for, and find their way to, a large audience. So far the results were only shown under non-disclosure conditions, so we cannot report on that yet. It is encouraging, however, that many viewers asked when the results would become publicly available and if they could be notified on this.

There is room for more future work. We are considering alternative methods to produce geography aligned meshes. An interesting option is to use a physically based model to simulate crack formation (Iben and O'Brien, 2006). Also, the methods presented here can be used for a variety of other purposes, for instance to show plate tectonics, voyages of discovery, or scientific data given for spherical

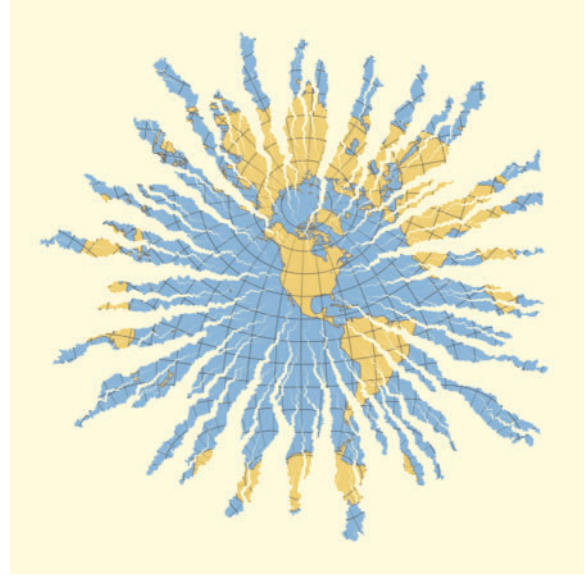


Figure 14. Azimuthal projection, random weights added, 81 920 polygons

angles. Such applications can be produced easily, just by varying the map used for mesh alignment.

BIOGRAPHICAL NOTE



Jarke J. van Wijk is a full professor of Visualisation at the Technische Universiteit Eindhoven. He has a MSc in Industrial Design and a PhD in Computer Science, both from the Delft University of Technology. His main research interests are Information Visualisation and Flow Visualisation, with a focus on developing novel and effective visual representations. He has been paper

co-chair of the IEEE Visualisation and IEEE Information Visualisation conferences.

ACKNOWLEDGEMENTS

The author thanks Michiel Wijers for coining the term 'myriahedral', and Jason Dykes and Menno-Jan Kraak for their encouragement and advice.

REFERENCES

- Alliez, P., Cohen-Steiner, D., Devillers, O., Levy, B. and Desbrun, M. (2003). 'Anisotropic polygonal remeshing', *ACM Transactions on Graphics* 22(3), 485–493. Proceedings SIGGRAPH 2003.
- Dutton, G. (1996). 'Encoding and handling geospatial data with hierarchical triangular meshes', in *Advances in GIS Research II*

- (**Proc. SDH7, Delft, Holland**), 505–518, ed. by Kraak, M.-J. and Molenaar, M., Taylor & Francis, London.
- Floater, M. S. and Hormann, K. (2005). 'Surface parameterisation: a tutorial and survey', in **Advances in multiresolution for geometric modelling**, 157–186, ed. by Dodgson, N. A., Floater, M. S. and Sabin, M. A. (eds), Springer Verlag.
- Fredman, M. L., Sedgewick, R., Sleator, D. D. and Tarjan, R. (1986). 'The pairing heap: A new form of self-adjusting heap', **Algorithmica** 1(1), 111–129.
- Furuti, C. A. (2006). 'Map Projections' <http://www.progonos.com/furuti/MapProj>.
- Helman, J. L. and Hesselink, L. (1991). 'Visualizing vector field topology in fluid flows', **IEEE Computer Graphics and Applications** 11(3), 36–46.
- Iben, H. N. and O'Brien, J. F. (2006). 'Generating surface crack patterns', in **Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2006**, Vienna, Austria, 177–185, ed. by O'Sullivan, C. and Pighin, F., Eurographics association, Aire-la-Ville, Switzerland.
- Jobard, B. and Lefer, W. (1997). 'Creating evenly-spaced streamlines of arbitrary density', in **Visualization in Scientific Computing '97**, 43–56, ed. by Lefer, W. and Grave, M., Springer Verlag.
- Karni, Z., Bogomjakov, A. and Gotsman, C. (2002). 'Efficient compression and rendering of multi-resolution meshes', **Proceedings of IEEE Visualization 2002**, 347–354, ed. by Moorhead, R., Gross, M. and Joy, K.I., IEEE Computer Society Press.
- Kraak, M.-J. and Ormeling, F. (2002). **Cartography: Visualization of Geospatial Data (2nd edition)**, Prentice Hall, London.
- Monmonier, M. (1991). **How to Lie with Maps**, University of Chicago Press, Chicago.
- Moret, B. M. E. and Shapiro, H. D. (1991). 'An empirical analysis of algorithms for constructing a minimum spanning tree', **Lecture Notes in Computer Science** 555, 192–203, Springer Verlag.
- Pape, D. (2001). 'Earth images', www.evl.uic.edu/pape/data/Earth.
- Prim, R. (1957). 'Shortest connection networks and some generalisations', **Bell System Technical Journal** 36, 1389–1401.
- Robinson, A. H., Morrison, J. L., Muehrcke, P. C. and Kimerling, A. J. (1995). **Elements of Cartography**, Wiley.
- Slocum, T. A., McMaster, R. B., Kessler, F. C. and Howard, H. H. (2003). **Thematic Cartography and Geographic Visualization, Second Edition**, Prentice Hall.
- Snyder, J. P. (1993). **Flattening the Earth: Two Thousand Years of Map Projections**, University of Chicago Press.
- Sorkine, O., Cohen-Or, D., Goldenthal, R. and Lischinski, D. (2002). 'Bounded-distortion piecewise mesh parameterisation', **Proceedings of IEEE Visualization 2002**, 355–362, ed. by Moorhead, R., Gross, M. and Joy, K.I., IEEE Computer Society Press.
- Spilhaus, A. (1983). 'World ocean maps: The proper places to interrupt', **Proceedings of the American Philosophical Society** 127(1), 50–60.