

Using Turtles and Skeletons to Display the Viewable Sphere

David Swart
Waterloo, Ontario, Canada
E-mail: david.swart@sympatico.ca

Abstract

A number of software packages allow us to stitch together photos to capture the entire viewable sphere. To represent the viewable sphere in interesting ways, we introduce a new way to define projections (cartographic mappings from the sphere to the plane) using a turtle language similar to Logo. These projections can be printed, cut out and assembled into photographic equivalents of Termespheres. Further, this new method for defining projections is versatile enough to allow artistic and mathematical explorations and as such, the projections can be interesting in their own right (i.e., without being constructed back into spheres).

1 Motivation

Image stitching software makes it easier for users to create larger, seamless images from photos. Panoramas that capture the entire viewable sphere can be stitched together, but how does one represent, display, and share them? To display these panoramas on a flat computer screen or on paper, we need a mapping from the sphere to the plane. In cartography, these mappings are referred to as *projections* and all projections will introduce some sort of distortion. Despite this, projecting the visible sphere onto the plane has been undertaken by photographers and mathematicians alike, and conformal mappings (mappings with no local distortions) play a big role [5]. But perhaps a more natural medium for representing panoramas would be the photographic equivalent of a *Termisphere* [6]. Dick Termes painted spherical representations of the viewable sphere so that rather than looking outward from the observer's viewpoint, the observer looks inward to the scene on the surface of the sphere. If a spherical representation is chosen, then the photographer's problem would then become one of manufacturability.

A common method for creating globes by gluing thin paper strips called *gores* onto a sphere can result in creases and wrinkles. More gores reduce the amount creases but make the task of assembly more finicky. Another solution is to include features akin to a dressmaker's *darts* where a thin triangle of material is removed and the edges are brought together to shape the sphere. Many commercial globes use this approach. Towards creating complex projections such as gores with darts, this paper describes a versatile approach to designing projections that are suitable for printing and assembling back into spheres.

The problem of covering a sphere with paper will be familiar to those who have seen the work by Demaine *et al.* [4]. Their solution uses the same fundamental mechanism as the one presented here (i.e., the *taut paths* in their terminology are called *skeletons* here). While they discuss metrics for area, perimeter, and tiling the plane, this paper introduces the new concepts of a) a turtle language for the purpose of designing projections more easily b) having a disconnected set of taut line segments c) a focus on mapping the interior imagery and d) an exploration of some different families of projections.

This paper is organized as follows: Section 2 introduces the concept of skeletons and how they map the sphere to a plane while Section 3 describes how we can use a turtle language to define these skeletons. Section 4 walks through an example. Section 5 discusses some tips on implementation. Finally, Section 6 displays some projections created with the new method.

2 Mapping the Sphere to the Plane

In keeping with the idea that we intend to paste the planar projection back onto a sphere, it helps to think of some lines and points on a ball where we glue or tack pieces of paper. And now, imagine pulling the unglued portions of paper taut and down onto the ball. In this scenario, the paper is the plane and the ball is the sphere of course. We call the set of line segments and points that correspond to the glue the *skeleton of the projection*.

The method for mapping the sphere to the plane becomes more apparent when we realize that this skeleton exists as a set of line segments on the sphere and a corresponding set on the plane. We still need to establish how to map the non-skeleton points of the sphere (mimicking pulling the unglued portion of the paper taut). But once we do so, we can define projections by specifying a skeleton as a set of line segments on the sphere, and a corresponding set of line segments with the same lengths on the plane.

Given some line segments on the sphere, some line segments on the plane, and given an arbitrary point on the sphere, the task is to find a point on the plane in the same relative position and distance to the segments on the plane. We define the mapping for both skeleton and non-skeleton points below. In this paper, distance on the sphere refers to great circle-distance (i.e., the distance along the surface).

Definition. Given S , a set of line segments on a sphere, and S' a corresponding set of line segments of the same lengths on the plane, we define a function $f_{S,S'}$ that maps the points on a sphere to the plane. Let p be a point on the sphere. Let segment $ab \in S$ be the segment nearest p , with corresponding segment $a'b' \in S'$. If p lies on ab , then we define $f_{S,S'}(p)$ as the point p' on $a'b'$ such that $|ap| = |a'p'|$. Otherwise, p

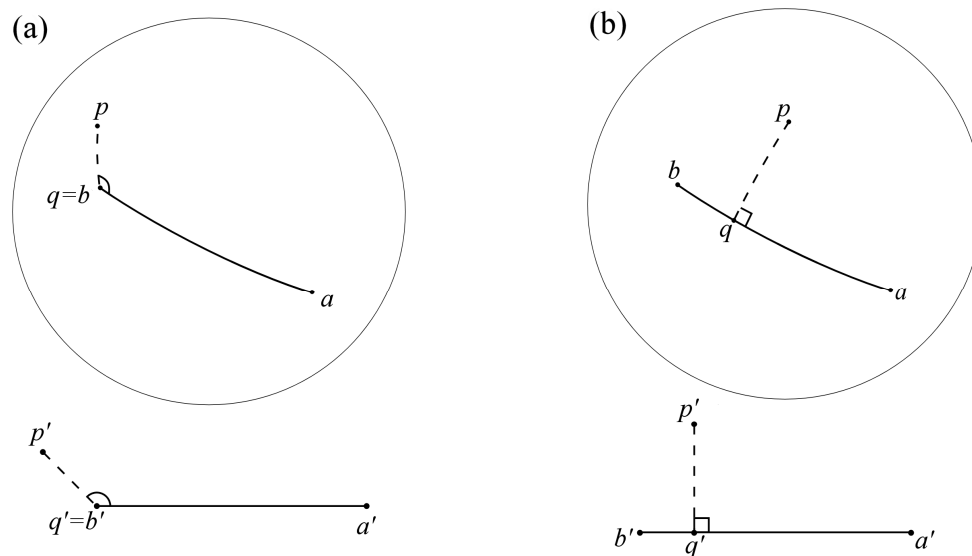


Figure 1: Mapping a point p on the sphere to a point p' on the plane based on the nearest segment to p and its corresponding segment on the plane. (a) depicts when p is closest to an endpoint of ab , (b) depicts when it is closest to a non-endpoint

does not lie on ab . So let q be the point on ab nearest to p (without loss of generality, $q \neq a$). Then let q' be the corresponding point on $a'b'$ such that $|aq|=|a'q'|$. We define the function $f_{s,s}(p)$ as the point p' on the plane, such that $\angle aqp = \angle a'q'p'$ and $|pq|=|p'q'|$.

The above function is not well defined if $|ab|=0$, i.e., in the degenerate case where the segment is just a point, because the required angle is not defined. However, we can apply $f_{s,s}$ to skeletons with points if we keep a sense of direction with each skeleton point.

In our intuitive picture of gluing pieces of paper to a ball, we left out the details of what shape the pieces of paper will be. The definition of $f_{s,s}$ defines the paper shapes when it requires that we must use the nearest segment to map our points. Thus the sphere is divided neatly into *Voronoi regions* (neighborhoods of the closest points) around each line segment. And hence the shapes of the pieces of paper that get glued to the ball are the shapes of these Voronoi regions, mapped around their corresponding skeleton line segments on the plane.

3 Defining a Skeleton with a Turtle Language

The previous section defined a mapping that uses a given skeleton to map the sphere to the plane. To define the skeleton itself, we can make use of the concept of a turtle language. This simplifies the definition of the skeleton line segments and also places the skeleton on the plane in a logical arrangement.

Turtle languages such as Logo are graphic programming languages that use commands to move a "turtle" around on the screen [1]. The turtle has a pen which draws a line along its path. In our situation we have two spaces, the plane and the sphere. It makes sense then to have two turtles, a sphere-turtle and a plane-turtle, which simultaneously obey the same set of instructions.

The list of the primitives that we use is in Table 1. Turtle language commands are usually simple primitives such as "pen up", "move forward 10", and "pen down" however our set of commands use *move-to* and *line-to* commands to accomplish the same thing. The units listed in the table are chosen for practical purposes; a line segment measuring π radians on the sphere corresponds to a line segment of length n pixels on the plane.

Table 1 *Commands in our turtle language.*

R x	Rotate in place $x \times \pi$ radians counter-clockwise
M x	Move $x \times \pi$ radians on the sphere, $x \times n$ pixels on the plane. No segment is created
L x	Create a line segment forward $x \times \pi$ radians on the sphere, $x \times n$ pixels on the plane.

Defining the skeleton with a turtle language this way has the benefit of arranging the skeleton on the plane in the right position and at the right distance, such that the mapped Voronoi regions are just touching and the imagery within adjacent regions is aligned.

Since the regions are well aligned by this method, designing a skeleton solely in "sphere-space" usually results in pleasing results on the plane. However, it is still possible to have a mapping where different points on a sphere get mapped to the same position on the plane! (An exercise for the reader is to peel a mandarin orange in one piece such that when flattened, the rind overlaps itself.) In practice, such situations are rare and minor enough to ignore. But if necessary, one could add an "M 2" instruction which returns the sphere pointer to its original position, but moves the plane pointer $2n$ pixels away.

4 The Volleyball Projection (A Demonstration)

Let us now illustrate the process of mapping the sphere to the plane based on the design of a volleyball's panels. Instead of a panorama, we use a map of the world for illustration purposes since it is familiar, and it shows up well in this medium. The sphere-turtle starts out in Banff, Alberta, Canada and faces east.

The projection process starts by instructing the sphere-turtle to mark the middles of the leather panels. See Figure 2(a). For instance, the specific turtle instructions for three adjacent panels on a side are:

```
M -1/6; L 1/3; M -1/6; R 1/2; M 1/6; R -1/2;
M -1/6; L 1/3; M -1/6; R 1/2; M -1/3; R -1/2;
M -1/6; L 1/3; M -1/6; R 1/2; M 1/6; R -1/2;
```

The sphere-turtle is moved to the other five sets of three panels and the above sequence is repeated. Simultaneously, the plane-turtle follows along and automatically creates an organized layout. See also Figure 2(a).

The drawn line segments of the turtles' path create the two skeletons used for the mapping, one skeleton on the sphere, and one skeleton on the plane. See Figure 2(b).

The points of the sphere are thus arranged into Voronoi regions around the skeleton's line segments, Figure 2(c). Finally, using $f_{s,s'}$, the points of each Voronoi region are mapped around the corresponding line segments on the plane to create the final volleyball projection. See also Figure 2(c).

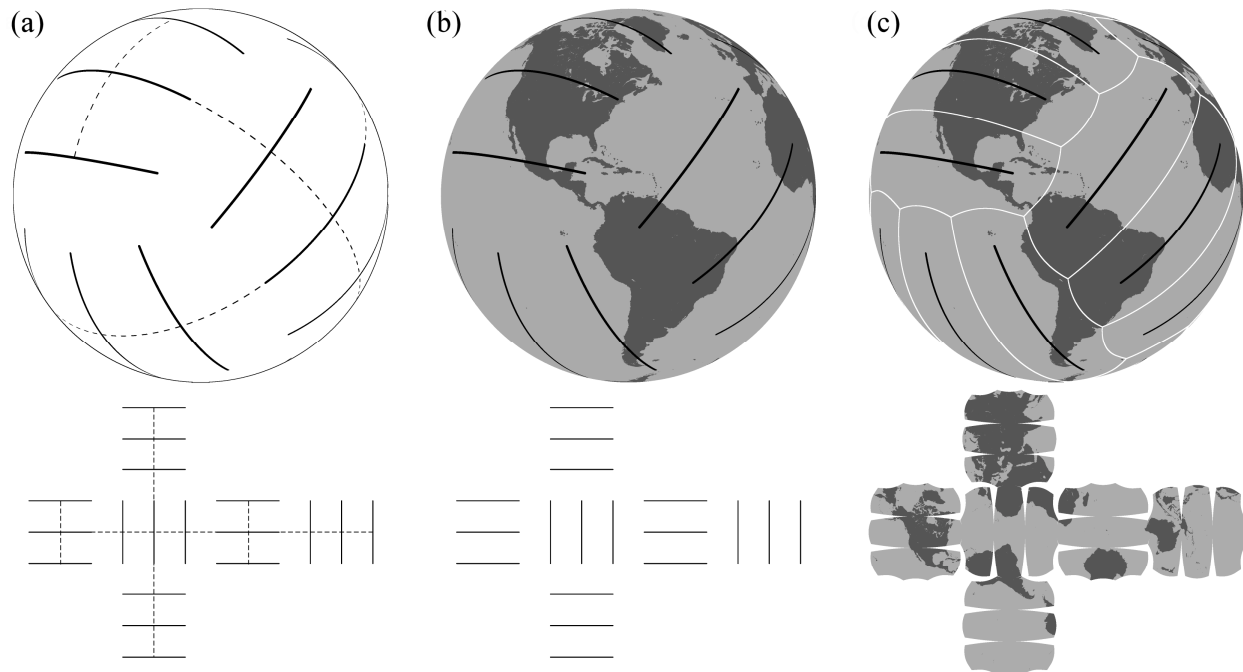


Figure 2: (a) A single turtle program directs two turtles to draw paths, one on a sphere and one on a plane. (b) Two skeletons (sets of line segments) are thus defined. (c) The points nearest to each line segment on the sphere (Voronoi regions) are mapped to the corresponding segment on the plane.

5 Implementation

The preceding section demonstrates some important concepts of our skeleton method but does not serve well as a guide to implementing an algorithm. To render an image, we might start with the position of

each pixel p' on the plane and then calculate the point p on the sphere to render. However, the function $f_{s,s'}$ maps points in the opposite direction; we would need to implement its inverse. And yet, $f_{s,s'}$ has *no* inverse in the rare cases when it maps two points on a sphere to the same point on the plane. Our implementation gets around this technicality by blindly ignoring it. (If necessary though, one could use the “M 2” work around described at the end of Section 3.)

The pseudocode below may help those wishing to write their own software. We define function $g_{s',s}$ in the same way that $f_{s,s'}$ is defined in Section 2, but with the roles of the sphere and the plane reversed. Thus when $f_{s,s'}(p)=p'$ is single-valued, then yes, $g_{s',s}(p')=p$, but only when p is in the proper Voronoi region on the sphere. The pseudo-code below applies $g_{s',s}$ and then performs this check.

```

The input is a skeleton and a color for each point  $p$  on the sphere
for each pixel  $p'$  on the plane, do:
   $s' \leftarrow$  nearest segment on the plane to  $p'$ 
   $s \leftarrow$  segment on sphere corresponding to  $s'$ 
   $p \leftarrow g_{s',s}(p')$  (the point on the sphere in the same relative position to  $s$  as  $p'$  is to  $s'$ )
  if the nearest segment to  $p \neq s$  then:
    output a blank pixel
  otherwise:
    output the color of  $p$ 

```

Figure 3: Pseudocode for one possible implementation

Our implementation is a Win32 console application written in C, with no graphical user interface. Its running time has not been formally analyzed but experience shows a strong dependency on the number of segments in the skeleton. For instance, it takes about five seconds to render a 2000x2000 image of a two-segment skeleton, but almost 30 minutes for a 600-segment skeleton of the same size.

6 Examples

We finish by looking at some of the projections accomplished with our new method.

6.1 Simple Projections Even very short turtle programs can generate some interesting results. A single point skeleton specified by “L 0” creates a projection known as an azimuthal equidistant projection. See Figure 4. A skeleton consisting of a great circle is specified by “L 2” and produces a well known and common projection known as an equirectangular projection. See Figure 5.



Figure 4: An azimuthal equidistant projection centred on Banff.

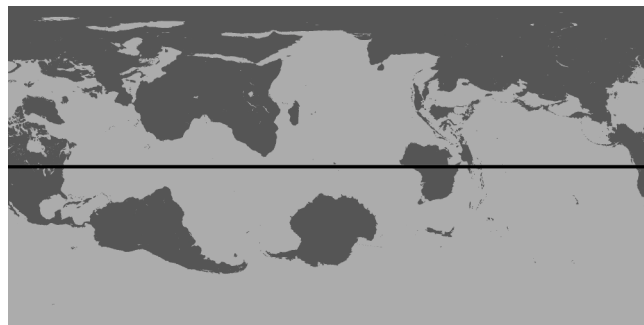


Figure 5: An equirectangular projection, with Banff at the middle of the left edge.

The previous two projections are notable since they help characterize the interior of any projections produced with our skeleton method. Mathematically, each projection is a piecewise mix of azimuthal equidistant projections and equirectangular projections depending upon the two cases shown in Figure 1.

6.2 Polyhedra. Skeletons consisting of the face centres of polyhedra projected onto the sphere can produce interesting projections. Such projections resemble unfolded and flattened polyhedra, although not exactly (for instance, the edges of the faces are curved). The pentagonal hexecontahedron based projection in Figure 6 is my personal favourite.

6.3 Sport Balls. We have already seen the volleyball. And we have already discussed polyhedra, but the soccerball needs a special mention. In this case, our polyhedron approach mentioned in Section 6.2 does not create regular hexagons: the distance between the face centres of two neighbouring hexagons is greater than between a hexagon and a pentagon, and thus the hexagons' Voronoi regions are not regular. A new feature was added to our turtle language to give a weighting to each skeleton segment. By scaling up the distance to each pentagon centre appropriately, we achieve our soccerball projection in Figure 7.

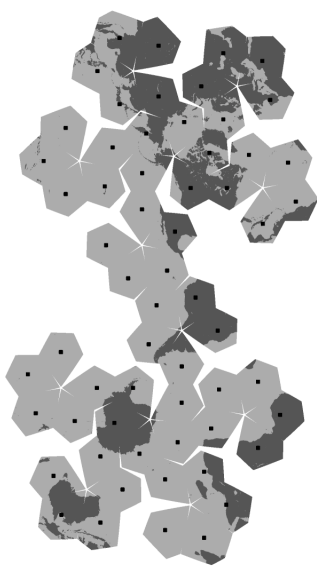


Figure 6: *A pentagonal hexecontahedron projection*

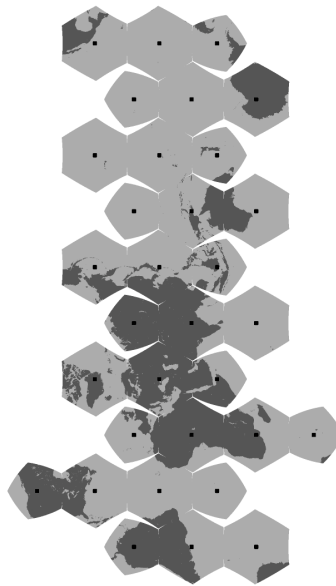


Figure 7: *A soccerball projection*

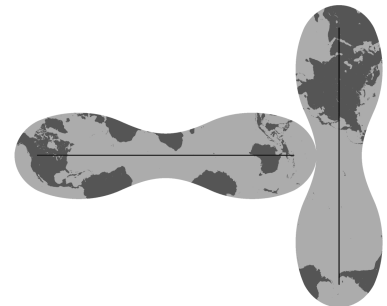


Figure 8: *A baseball projection*

As for baseballs, John Conway once conjectured, “no two definitions of “the correct baseball curve” will give the same answer unless their equivalence is obvious from the start” [3]. The set of turtle instructions “L $2x$; M $1-x$; R $1/2$; M $-x$; L $2x$;” with $x=0.739726$ throws another definition into the mix. See Figure 8. The value of x was chosen such that the leather flats are $1^{3/16}$ inches wide at the “waist” for a baseball with a circumference of $9^{1/8}$ inches, according to measured values.

6.4 Trail of Small Segments. Lloyd Burchill of Flaming Pear Software created a projection which mimics the peeling of an apple [2]. We can accomplish the same effect by specifying a trail of one hundred or so line segments which spiral from the top to the bottom of the “apple”. See Figure 9.

6.5 Gores. The standard layout of gores can be specified with a line segment from pole to pole for each gore. This is well explored in [4]. Dart features are created by adding gaps to the meridians (Figure 10).

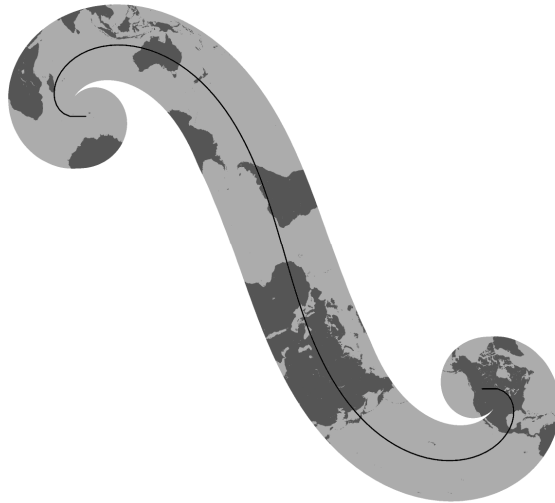


Figure 9: A rind projection, after a projection by Lloyd Burchill

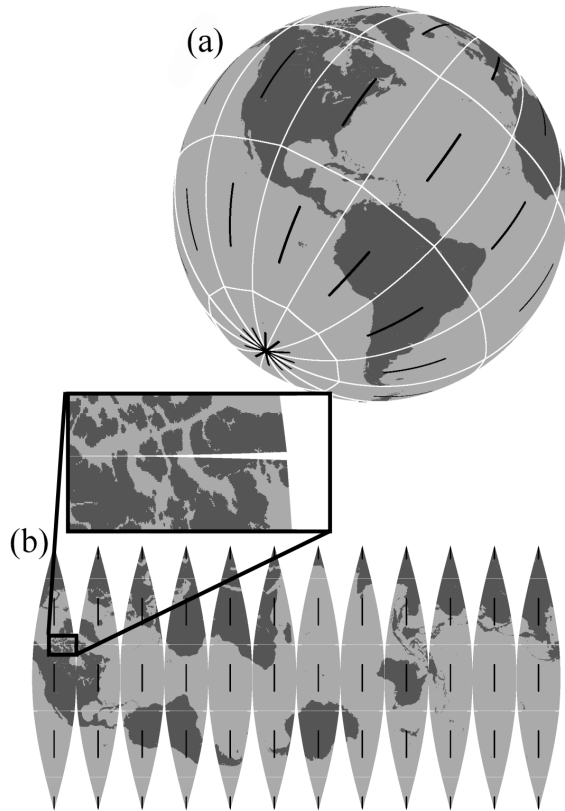


Figure 10: The skeleton to create gores with darts shown on (a) the sphere and (b) the plane. A dart is shown inset.

6.6 Photographs. Returning to the original reason for this work, Figure 11 shows a constructed photographic Termesphere using the gore design shown in Figure 10. The paper is cut and glued over a 3" plastic ball and used as a Christmas tree ornament.

We finish by showing how our new projections can be used as a compositional element for panoramas. Figure 12 shows four more projection designs.

7 Acknowledgements

I would like to acknowledge the inspiring work of Lloyd Burchill and Sébastien Pérez-Duarte. I also thank them, Michael Swart, Judy Newell and the anonymous reviewers for their feedback. The source imagery of the earth is courtesy of NASA's Earth Observatory. Figures 11 and 12 are © David Swart. The source imagery for Figure 12(c) is courtesy Sébastien Pérez-Duarte and used with permission.



Figure 11: Photographic Termesphere

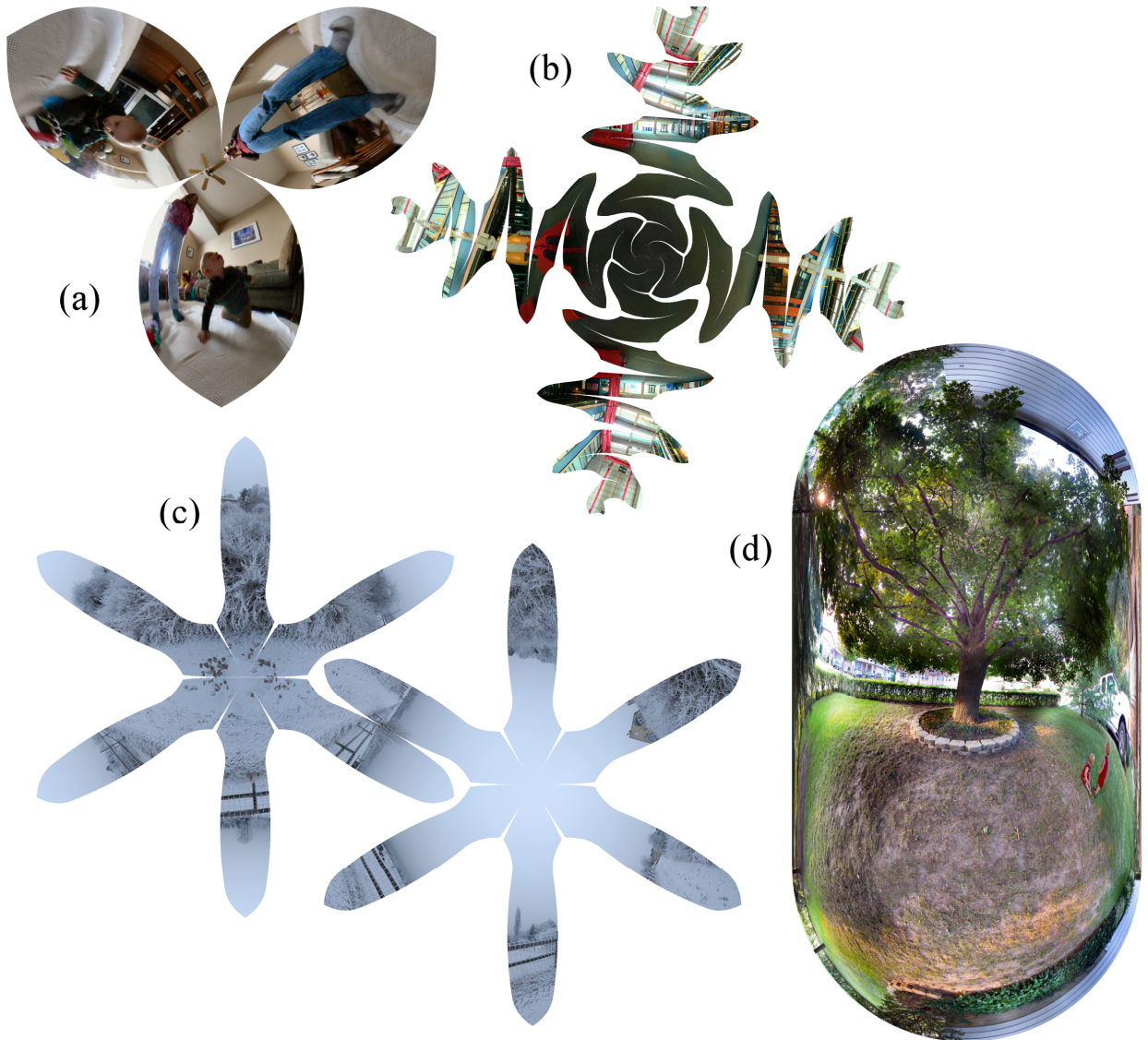


Figure 12: “Twin’s Eye View”(a) uses a three-gore projection as a compositional element. “Meandering Meridians”(b) and “Snowflakes on Snowflakes”(c) explore the potential of the turtle method with less emphasis on their panoramic content. “Under the Maple Tree”(d) is produced by the turtle instruction “L 1”.

References

- [1] Harold Abelson and Andrea diSessa, *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. MIT Press, Cambridge. 1981.
- [2] Lloyd Burchill. English Bay Park. <http://www.flickr.com/photos/lloydb/57038900> 28 Oct 2005.
- [3] John Conway. math-fun. <http://www.math.uni-bielefeld.de/~sillke/PUZZLES/baseball> 3 Jun 1994.
- [4] Erik D. Demaine, Martin L. Demaine, John Iacono, and Stefan Langerman: Wrapping Spheres with Flat Paper, In *Computational Geometry: Theory and Application*, to appear. Special issue of selected papers from the 20th European Workshop on Computational Geometry. 2007.
- [5] Daniel M. Germán, Lloyd Burchill, Alexandre Duret-Lutz, Sébastien Pérez-Duarte, Emmanuel Pérez-Duarte, Josh Sommers: Flattening the Viewable Sphere. In *Computational Aesthetics 2007*, pages 23-28. 2007.
- [6] Dick Termes. Painting the Total Picture. In *Bridges Leeuwarden: Mathematical Connections in Art, Music and Science Proceedings 2008*, pages 363-368. 2008.