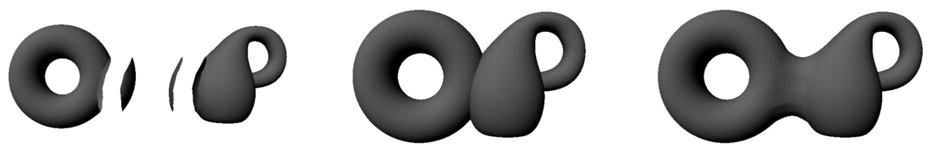
THESIS

1. Introduction

1.1. **Overview.** A connected sum of manifolds is made by removing a disc from each manifold and gluing the manifolds along the boundary.

My program is designed to use an intermediate manifold to allow them to be connected smoothly.



(a) Two surfaces with discs removed (b) Two spaces naively connected (c) Two spaces smoothly connected

Figure 1. Connecting two surfaces

???

I have written a graphics program to visualize three-dimensional non-Euclidean geometry, with emphasis on connected sums of manifolds.

I have implemented $\mathbb{E}^3$ and a space homeomorphic to $S^2 \times \mathbb{R}$, which I have called Wormhole.

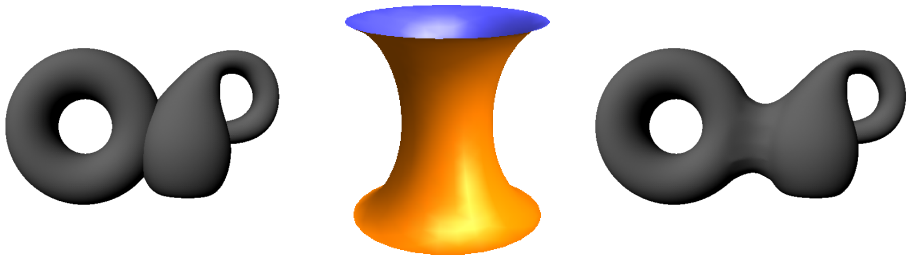
I have also implemented a method to glue multiple spaces together along spherical boundaries.

Outside Euclidean geometry, gluing spaces together like this won't always be smooth. More precisely, since two spheres with the same surface area do not generally have the same curvature, gluing the spaces together by those spheres will result in them having different curvature from each side. A similar problem will occur if the insides or outsides of two spheres are glued together.

In order to address this problem, I have designed a manifold I call the wormhole.

I define 2-Wormhole as a quotient space on $\mathbb{H}^2$ made by identifying two ultraparallel lines. I have designed 3-Wormhole as a higher-dimensional analogue of a 2-wormhole.

One obstacle was finding geodesics between two given points when spaces are glued together. Since light follows a geodesic, this is necessary to show where you'd see the point. In E^3 and Wormhole, it is not difficult to construct a geodesic between two given points. However, this does not apply to their connected sums.



(a) Two spaces naively connected (b) A wormhole (c) Two spaces smoothly connected

Figure 2

I can easily construct a geodesic from a given point moving in a given direction. By repeatedly constructing such geodesics at set lengths and taking note of the error, I can use Newton's method to solve numerically for the geodesic between two given points.

My program can be run either through rasterization or ray-tracing.

When used for rasterization, the program finds a geodesic from the camera to each vertex, finds the direction the geodesic moves from the camera, and draws the vertex in that direction. Then it draws triangles between those vertices.

When used for ray-tracing, it constructs geodesics corresponding to each pixel in the camera, and shoots rays to see what lies in that direction.

1.2. **Background.**

The prime decomposition theorem states that a compact, connected, orientable 3-manifold can be decomposed into a connected sum of prime manifolds (manifolds that cannot be decomposed further, except by trivially removing a sphere) [11], and that this is unique up to insertion or deletion of copies of S^3 [12]. The connected sum of two spaces is made by removing a ball from each and identifying their bounding spheres.

The torus decomposition theorem states that there is a minimal collection of disjointly embedded incompressible tori such that cutting along each yields components that are each either atoroidal or Seifert-fibered. It also states that this collection is unique up to isomorphism [4] [5] [6] [8].

The geometrization theorem states that any 3-manifold can be decomposed canonically into submanifolds which each is a quotient space of one of the following eight geometries: $S^3, \mathbb{E}^3, \mathbb{H}^3, S^2 \times \mathbb{R}, \mathbb{H}^2 \times \mathbb{R}, \tilde{SL}(2, \mathbb{R})$, Nil geometry, and Sol geometry. This is done by decomposing along the spheres given by the prime decomposition theorem and tori given by the torus decomposition theorem. The initial work was done by Grisha Perelman [14] [15] [16], and was later completed by other mathematicians [10] [1] [7].

My program allows the smooth creation of a connected sum of Euclidean geometries as in the geometrization theorem, and visualization of the result. My program can also be easily expanded to include hyperbolic and spherical geometry. "Smoothness" means that the boundaries that are identified have the same curvatures in each surface. If they are not smoothly identified, then they will appear to have different curvatures from each side. As a result, a geodesic that barely intersects the border can have a very different path from one that barely misses.

For example, if we glue the outside of a sphere in Euclidean geometry to the outside of another such sphere in another copy of Euclidean geometry, the result looks like the portal is a reflective sphere, but with the reflection from the other geometry. This has effects such as blocking anything behind the sphere. This is impossible in a true manifold. Any point is visible from any other. My program avoids this by using an intermediate geometry of non-constant curvature which contains spheres that can be glued to spaces of any constant curvature.

My program does not support torus decompositions, and cannot show every manifold. However, it is still more general than anything that has been reported previously, as described next.

Previous work has been done to visualize S^3 and $\mathbb{H}^3$ and quotient spaces thereof. In particular, Weeks has written a program that can view compact quotient spaces of $S^3, \mathbb{E}^3$, and $\mathbb{H}^3$ [19]. He also wrote a paper that describes the process in depth [20].

Gunn and Maxwell made a video showing the complementary spaces of knots, most of which are quotient spaces of $\mathbb{H}^3$ [3]. Gunn explains the techniques he uses in [2] [17].

Levy, Munzner and Mark Phillips wrote a geometry viewer called Geomview. Among its features is the ability to render non-Euclidean geometry [18].

The main difference that distinguishes my work is the ability to create connected sums between spaces.

I am unaware of any previous attempts to visualize general connected sums of geometries. However, attempts have been made to visualize a connected sum of two $\mathbb{E}^3$ spaces, with curvature near zero outside of a small wormhole.

Rune Johansen made a video that was designed to convey the idea of a wormhole [9]. This video is primarily artistic in nature, and uses various tricks to make space look curved. There is no geometry that looks precisely like the video.

Corvin Zahn made a computer generated video precisely illustrating a wormhole [21]. He uses a solution to Einstein's field equations that had been found previously [13], and simulated the camera moving through this wormhole.

Zahn's wormhole was made as one continuous geometry with everywhere negative curvature. While this is a much more interesting space than what my program will make, it's much harder to generalize. Adding a second wormhole between the spaces or a second wormhole leading to a third space would require redesigning the entire manifold. In contrast, my program provides a compact manifold with boundary to connect spaces, so as long as the spheres they replace in Euclidean geometry do not touch, any number can be added easily to the same space.

In adddition, Zahn seems to have used a ray-tracing method and found the geodesics numerically. My program can use rasterization to run at near real-time, or solve geodesics symbolically to speed up ray-tracing.

1.3. **Wormhole.** In order to allow more interesting manifolds than a few that I hard code in, I have implemented connected sums of manifolds. Unfortunately, almost none of the well-known manifolds can be smoothly connected without changing their metrics. In order to facilitate this, I have found a class of manifolds that can be used as intermediates to smoothly connect two manifolds of constant curvature, and which allows geodesics to be easily calculated. I call them wormholes.

The wormhole is topologically $S^2 \times \mathbb{R}$, but with a different metric. A geodesic between two points on this metric can be found with the following method:

Given $p_1, p_2 \in S^2 \times \mathbb{R}, p_1 = (s_1, r_1), p_2 = (s_2, r_2)$ where $s_1, s_2 \in S^2, r_1, r_2 \in \mathbb{R}$. Let S be the great circle containing s_1, s_2. The points p_1, p_2 are in the slice $S \times \mathbb{R}$ of $S^2 \times \mathbb{R}$. Let $\theta_1, \theta_2 \in S$ be s_1, s_2 under the inverse inclusion map. Consider $p'_1 = (\theta_1, r_1), p'_2 = (\theta_2, r_2)$. $S \times \mathbb{R}$ corresponds to a quotient space of the half-plane model of $\mathbb{H}^2$ via $\phi : (\theta, r) \mapsto (e^{k\theta} \cos r, e^{k\theta} \sin r)$ where k is a constant depending on the quotient space. The two semicircles in figure 1.3 are identified to form the quotient.

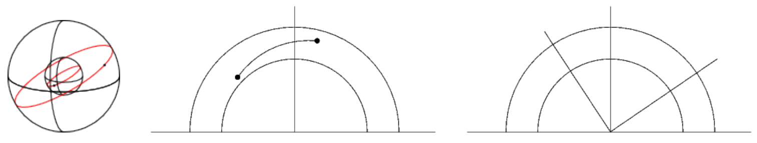

(a) $S^2 \times \mathbb{R}$ (b) Half Plane (c) Radial Lines

Figure 3. Illustration of finding geodesics on Wormhole

Find a geodesic in $\mathbb{H}^2$ that connects $\phi(p'_1)$ and $\phi(p'_2)$, and map it back to $S \times \mathbb{R}$ and finally $S^2 \times \mathbb{R}$.

Radial lines in $\mathbb{H}^2$ are curves of constant curvature. These map to closed loops in $S \times \mathbb{R}$. The angle of the line controls how the curvature and diameter relate, allowing connection to a space of a specific curvature, such as Euclidean or hyperbolic.

2. Details common to all spaces

2.1. **Exponential Map.** In a Euclidean graphics program, each point is drawn based on the displacement from the camera. Manifolds in general don't have a nice linear structure, so we need to be more clear as to exactly what is going on.

Each point emits photons that follow geodesics before reaching the camera and being detected. The camera draws the point based on the direction the photon is moving when it reaches the camera.

When dealing with Euclidean geometry, a direction can be thought of on its own. This doesn't work on manifolds in general. A direction can only be defined as at a specific point. The directions, along with distances can be used to build vector spaces at each point.

A vector is often defined on a manifold essentially as a direction of motion at a given point along with a magnitude. A camera would not have any way of knowing anything beyond the direction a photon is moving, but it will make the math easier to pretend that it knows the distance it travelled as well. The vector space of all the vectors at a given point $p \in M$ is referred to as M_p.

I should be clear that normal methods of finding distance, such as parallax and inverse square law, do not work on arbitrary manifolds. The ability for the camera to find distance is nothing more than a mathematical abstraction.

The map $\exp_p : M_p \to M$ where $\exp_p(\mathbf{v})$ is the endpoint of a geodesic starting at p, moving in the direction of $\mathbf{v}$ of length $\|\mathbf{v}\|$ is known as the exponential map.

Given a camera position $p \in M$, and a point $q \in M$, the camera does not see q. It only sees $\exp_p^{-1}(q)$, the direction and distance to q.

2.2. Finding the Geodesic Between Two Points Numerically.

In many manifolds, the geodesic between two points can be easily calculated symbolically. However, when two or more simple manifolds are glued together, this method quickly becomes infeasible. You would likely have to find a new equation for every combination, and it's likely that the final equation will quickly get too complicated to be solved easily.

Given a point, p, and a vector, $\mathbf{v}$, it's still fairly easy to find the geodesic that extends distance $\|\mathbf{v}\|$ from point p in the direction of $\mathbf{v}$, and by extension its endpoint $\exp_p(\mathbf{v})$: Find the corresponding geodesic in the sub-manifold that p is in. Find the nearest intersection with a wormhole, if one exists. Send the point of intersection, along with the orientation and the remaining vector, to the other side of the wormhole in another sub-manifold. Extend the geodesic from there. Repeat until you reach the end of the geodesic.

There is not necessarily only one geodesic between a given pair of points. However, when drawing a triangle, the necessary geodesics will presumably be close to each other. This can be used to find the geodesic you're looking for as follows:

When drawing a triangle in which the geodesic reaching one of the vertices is known, the geodesic can be used for the first iteration to find the other two vertices.

For each iteration, we have a starting point p, ending point q, and a vector $\mathbf{v}$ which is the initial vector in the first iteration, and the result of the previous iteration otherwise. We are attempting to find the value of $\exp_p^{-1}(q)$ that's close to $\mathbf{v}$.

Let $\phi : M \to \mathbb{R}^3$ be an arbitrary map that is locally continuous and injective in a neighborhood of q.

Now we have $\phi \circ \exp_p : \mathbb{R}^3 \to \mathbb{R}^3$. This can be inverted with Newton's method. Then find $(\phi \circ \exp_p)^{-1} \circ \phi(q) = \exp_p^{-1} \circ \phi^{-1} \circ \phi(q) = \exp_p^{-1}(q)$. As long as the triangles are sufficiently small, $\mathbf{v}$ will be sufficiently close to the preferred solution, guaranteeing that Newton's method converges on that solution.

There are a few special cases where the above method won't be sufficient. I have come up with methods to solve these cases, but they have not been implemented. However, one case doesn't come up with the spaces that have been implemented ($\mathbb{E}^3$ and Wormhole), and the other is only necessary to draw multiple images of each triangle. The program draws only one image, but the image drawn is still drawn correctly.

2.3. **Orientation.** The camera sees everything as $\mathbb{R}^3$, with x, y and z coordinates. I will refer to these as the camera coordinates. The reason that we use $\mathbb{R}^3$, and not the geometry we are inside is that the camera does not directly see how space is warped. It only knows how far away points are, and in which direction. (Realistically, the camera only knows direction, but it simplifies calculations to pretend that it also knows distance.) This is referred to as $\mathbb{R}^3$ and not $\mathbb{E}^3$ because it has an inherent coordinate system, as opposed to simply following Euclidean geometry.

A camera must have an orientation. The orientation is essentially a linear isometry between the camera coordinates and the tangent space of the manifold at that point. In the case of $\mathbb{E}^3$, we can just give the space coordinates, resulting in a natural map from the coordinate space $\mathbb{R}^3$ to the camera coordinates $\mathbb{R}^3$, so the orientation can be thought of as a rotation from $\mathbb{R}^3$ to itself. Unfortunately, this strategy does not generally work, because manifolds in general cannot be given linear coordinate systems.

In order to store the orientation, I use an implicit default orientation for each point. If the coordinate map is conformal, a convenient default orientation is the differential of the coordinate map from $\mathbb{R}^3$ to the manifold. For example, in the half-plane model of $\mathbb{H}^3$, the vector $(0, 0, 1)$ maps to the direction of the geodesic that approaches $(0, 0, \infty)$.

Unfortunately, the coordinate map is not always conformal. Currently, the only exception is Wormhole, which maps from $S^2 \times \mathbb{R}$ instead of $\mathbb{R}^3$. In this case, $(0, 0, 0, 1)$ maps along the $\mathbb{R}$ axis and $(x, y, z, 0)$ maps to the tangents to the spherical cross-section in the obvious way. For example, $(1, 0, 0, 0)$ maps to a vector where the x component of the S^2 component is increasing. Everything else maps as necessary to make it angle-preserving.

???

For a manifold M and a point p, M_p refers to the tangent plane of M at p. This is isomorphic to $\mathbb{R}^3$.

Let $D_p : \mathbb{R}^3 \to M_p$ refer to the default orientation at point p on manifold M. It is a linear isometry that carries an explicit vector of the form (x, y, z) to the tangent plane M_p.

Given some orientation $O : \mathbb{R}^3 \to M_p$, I can store $D_p^{-1}O : \mathbb{R}^3 \to \mathbb{R}^3$.

If I want to rotate the camera by R, I am changing the orientation to OR. Since this is stored as $D_p^{-1}OR$, I can just post-multiply $D_p^{-1}O$ by R.

The parallel transport from p to q induces a transformation $P : M_p \to M_q$. When a point of reference is moved across such a path, the final orientation is $PO : \mathbb{R}^3 \to M_q$. This is stored as $D_q^{-1}PO$. Since it was originally stored as $D_p^{-1}O$, this is equivalent to premultiplying it by $D_q^{-1}PD_p$.

???

Everything above is written in 3 dimensional coordinates. As such, it doesn't quite apply for Wormhole, which uses a 4-dimensional coordinate system.

Wormhole is stored as $S^2 \times \mathbb{R} \subseteq \mathbb{R}^4$. The orientation is $\mathbb{R}^4 \to T_p$ where $p \in \mathbb{R}^4$. It also is designed to map $\langle e_1, e_2, e_3 \rangle$ to the subspace of T_p tangent to $S^2 \times \mathbb{R}$, and $\langle e_4 \rangle \mapsto N_p$, where N_p is the subspace of T_p normal to the tangent space of $S^2 \times \mathbb{R}$. In other words, e_4 maps to a vector perpendicular to reality.

Camera coordinates are transformed to $\mathbb{R}^4$ by adding a zero coordinate at the end. $\mathbb{R}^4$ is turned to camera coordinates by removing the trailing zero. If the trailing coordinate is not zero, the vector is not tangent along reality, and something clearly went wrong.

If I want to rotate the camera by the 3×3 matrix M, I am only rotating the first three coordinates. Those that stay within reality, so I can just multiply by $\begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}$, which rotates the first three coordinates that way and fixes the coordinate that lies perpendicular to reality.

???

My program does have objects besides cameras that have orientations. I have referred to the object as PointOfReference in the program itself. The orientation works the same, even if it isn't strictly speaking a camera.

3. Details on Specific Spaces

3.1. **2d Hyperbolic Geometry.**

3.1.1. *Finding the direction and distance from one point to another.* We will be using the half plane model: $\mathbf{H}^2 \mapsto \{(x_1, x_2) : x_2 > 0\}$.

Given points $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y} = (y_1, y_2)$, we need to find the vector $\mathbf{v}$ at $\mathbf{x}$ that is tangent to the geodesic from $\mathbf{x}$ to $\mathbf{y}$, such that $\|\mathbf{v}\|$ is the length of the geodesic arc between those points.

Assuming $x_1 \neq y_1$, the geodesic arc between $\mathbf{x}$ and $\mathbf{y}$ is mapped to an arc of a circle on the half-plane with its center on the x-axis. Consider this circle.

Let $\mathbf{c} = (c_1, c_2)$ be the center of the circle, and r be the radius. We know $c_2 = 0, \|\mathbf{x} - \mathbf{c}\| = \|\mathbf{y} - \mathbf{c}\| = r$.

First, let us solve for c_1.

We have

$$(x_1 - c_1)^2 + {x_2}^2 = (y_1 - c_1)^2 + {y_2}^2 = r^2.$$

Solving for

$$\begin{aligned} c_1 &= \frac{({x_1}^2 + {x_2}^2) - ({y_1}^2 + {y_2}^2)}{2(x_1 - y_1)} \\ &= \frac{\|\mathbf{x}\|^2 - \|\mathbf{y}\|^2}{2(x_1 - y_1)} \end{aligned}$$

We can use this value of c_1 to compute r: $r = \sqrt{(x_1 - c_1)^2 + x_2^2}$

The vector $\mathbf{v}$ is tangent to the circle, which means that it is at a right angle to the direction to $\mathbf{c}$, which works out to be $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (\mathbf{c} - \mathbf{x}) = \begin{pmatrix} x_2 \\ c_1 - x_1 \end{pmatrix}$. We then normalize this to $\dfrac{(x_2, c_1 - x_1)}{\sqrt{x_2^2 + (c_1 - x_1)^2}}$.

The distance along the geodesic can be calculated using the angle of the arc between $\mathbf{x}$ and $\mathbf{y}$ on the half plane. Assuming $\mathbf{y}$ is counterclockwise of $\mathbf{x}$, the length of the geodesic arc is:

$$\begin{aligned} &\int_{\theta_1}^{\theta_2} \frac{\sqrt{(\frac{d}{d\theta} r \sin\theta)^2 + (\frac{d}{d\theta} r \cos\theta)^2}}{r \sin\theta} d\theta \\ &= \int_{\theta_1}^{\theta_2} \frac{\sqrt{(r\cos\theta)^2 + (-r\sin\theta)^2}}{r \sin\theta} d\theta \\ &= \int_{\theta_1}^{\theta_2} \frac{r}{r\sin\theta} d\theta \\ &= \int_{\theta_1}^{\theta_2} \csc\theta d\theta \\ &= -\ln\left|\frac{\csc\theta_2 + \cot\theta_2}{\csc\theta_1 + \cot\theta_1}\right| \\ &= \ln\left|\frac{\csc\theta_1 + \cot\theta_1}{\csc\theta_2 + \cot\theta_2}\right|. \end{aligned}$$

We can substitute

$$\csc\theta_1 = \frac{r}{x_2}, \cot\theta_1 = \frac{x_1 - c_1}{x_2}, \csc\theta_2 = \frac{r}{y_2}, \cot\theta_2 = \frac{y_1 - c_1}{y_2}.$$

Hence, the distance is

$$\ln\left|\frac{\frac{r}{x_2}+\frac{x_1-c_1}{x_2}}{\frac{r}{y_2}+\frac{y_1-c_1}{y_2}}\right| = \ln\left|\frac{\frac{r+x_1-c_1}{x_2}}{\frac{r+y_1-c_1}{y_2}}\right| = \ln\left|\frac{y_2(r+x_1-c_1)}{x_2(r+y_1-c_1)}\right|$$

We already know x_2 and y_2 are greater than 0. Since $\|\mathbf{c}-\mathbf{x}\| = \|\mathbf{c}-\mathbf{y}\| = r$, clearly $|c_1 - x_1| < r$ and $|c_1 - y_1| < r$, which means $c_1 - x_1 < r$ and $c_1 - y_1 < r$, so $r + x_1 - c_1 > 0$ and $r + y_1 - c_1 > 0$. So $\frac{y_2(r+x_1-c_1)}{x_2(r+y_1-c_1)} \geq 0$. Thus, we may remove the absolute value, and the distance is equal to $\ln\frac{y_2(r+x_1-c_1)}{x_2(r+y_1-c_1)}$

If the path is clockwise instead of counterclockwise, the integral given is opposite the direction of motion, so the result is negative of the distance:

$$-\ln\frac{y_2(r+x_1-c_1)}{x_2(r+y_1-c_1)}.$$

In general, the distance is

$$\left|\ln\frac{y_2(r+x_1-c_1)}{x_2(r+y_1-c_1)}\right|.$$

If $x_1 = y_1$, the geodesic from $\mathbf{x}$ to $\mathbf{y}$ is a vertical line, so the problem is easier. The direction of $\mathbf{v}$ is $(0,1)$, and the distance is

$$\int_{x_2}^{y_2}\frac{1}{t}dt = \ln\frac{y_2}{x_2}.$$

This gives a vector of

$$\left(0, \ln\frac{y_2}{x_2}\right).$$

3.1.2. *Finding the point a given distance in a given direction from another.* Given initial point $\mathbf{x} = (x_1, x_2)$ and vector $\mathbf{v} = (v_1, v_2)$, we must find the endpoint $\mathbf{y}$ of the geodesic arc starting at $\mathbf{x}$ of length $\|\mathbf{v}\|$ in the direction of $\mathbf{v}$.

Suppose $v_1 \neq 0$. This means that the geodesic is an arc of a circle, instead of a vertical line segment. The center of the circle is on a line perpendicular to the tangent vector.

The Euclidean line tangent to the circle is $\mathbf{x} + t\mathbf{v}$, so the center of the circle is on

$$\mathbf{x} + t\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\mathbf{v} = (x_1 - tv_2, x_2 + tv_1).$$

This line intersects with the x axis when $x_2 + tv_1 = 0$.

Solving for t,

$$t = -\frac{x_2}{v_1}.$$

This implies
$$c_1 = x_1 - tv_2 = x_1 + \frac{x_2 v_2}{v_1},$$
and as before, $c_2 = 0$.

Now we can find $r = \|\mathbf{x} - \mathbf{c}\| = \sqrt{(x_1 - c_1)^2 + x_2^2}$.

Now that we have the circle, it's just a matter of finding the point at the right distance.

From our calculation above, we know distance,
$$d = \ln \frac{y_2(r + x_1 - c_1)}{x_2(r + y_1 - c_1)}.$$
$$e^d = \frac{y_2(r + x_1 - c_1)}{x_2(r + y_1 - c_1)}$$
$$x_2(r + y_1 - c_1)e^d = y_2(r + x_1 - c_1)$$
$${x_2}^2(r + y_1 - c_1)^2 e^{2d} = {y_2}^2(r + x_1 - c_1)^2$$
From the equation of a circle, we know
$$(y_1 - c_1)^2 + {y_2}^2 = r^2,$$
so
$${y_2}^2 = r^2 - (y_1 - c_1)^2.$$
Substituting this in,
$${y_2}^2(r + x_1 - c_1)^2 = [r^2 - (y_1 - c_1)^2](r + x_1 - c_1)^2$$
$$= [r - (y_1 - c_1)][r + (y_1 - c_1)](r + x_1 - c_1)^2$$
$$= (r - y_1 + c_1)(r + y_1 - c_1)(r + x_1 - c_1)^2.$$
Since $y_0 > 0$ and $r = \|\mathbf{y} - \mathbf{c}\|$, it must be the case that $|y_1 - c_1| < r$ so $r + y_1 - c_1 \neq 0$ and can be safely cancelled out of the equation
$${x_2}^2(r + y_1 - c_1)^2 e^{2d} = (r - y_1 + c_1)(r + y_1 - c_1)(r + x_1 - c_1)^2.$$
We obtain
$${x_2}^2(r + y_1 - c_1)e^{2d} = (r - y_1 + c_1)(r + x_1 - c_1)^2.$$
Solving for y_1, we get
$$y_1[{x_2}^2 e^{2d} + (r + x_1 - c_1)^2] = (r + c_1)(r + x_1 - c_1)^2 - {x_2}^2(r - c_1)e^{2d},$$
so
$$y_1 = \frac{(r + c_1)(r + x_1 - c_1)^2 - {x_2}^2(r - c_1)e^{2d}}{{x_2}^2 e^{2d} + (r + x_1 - c_1)^2}.$$
Now that we know y_1, we can easily find y_2 with $y_2 = \sqrt{r^2 - (y_1 - c_1)^2}$.

We will also need to find the change in orientation when moving along this geodesic as a parallel transport.

Let θ_0 be the angle of the normal to the geodesic at $\mathbf{x}$ with respect to half plane coordinates.

Now we need to parallel transport this normal along the geodesic to $\mathbf{y}$, and let θ_1 be the angle of the result with respect to half plane coordinates.

Since we are performing a parallel transport across a geodesic, the parallel transport of a normal is also normal to the geodesic, but at $\mathbf{y}$ instead of $\mathbf{x}$.

Note that since the geodesic is a circle, the coordinates for the normal vector are the same as the position of $\mathbf{x}$ relative to the center of the circle. This makes it easier to compute the angles.

First, let us compute $\sin\theta_0$ and $\cos\theta_0$ in terms of $\mathbf{x}, \mathbf{c}$, and r.

$$\sin\theta_0 = \frac{x_2 - c_2}{r} = \frac{x_2}{r}$$

$$\cos\theta_0 = \frac{x_1 - c_1}{r}$$

Similarly,

$$\sin\theta_1 = \frac{y_2}{r}, \cos\theta_1 = \frac{y_1 - c_1}{r}.$$

In order to computer a rotation, we need a rotation matrix which is computed with $\sin\Delta\theta$ and $\cos\Delta\theta$. Rather than solving for θ_0 and θ_1 with trigonometry, and finding the sin and the cosine of the difference, we can decrease the necessary computations by finding $\sin\Delta\theta$ and $\cos\Delta\theta$ with angle sums.

$$\sin(\theta_1 - \theta_0) = \sin\theta_1\cos\theta_0 - \cos\theta_1\sin\theta_1$$

$$\cos(\theta_1 - \theta_0) = \cos\theta_0\cos\theta_1 - \sin\theta_0\sin\theta_1$$

We use this to build the rotation matrix

$$\begin{pmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{pmatrix},$$

which was what we wanted.

If $v_1 = 0$ and the geodesic is a vertical line, so the vector $\mathbf{v}$ is pointing straight up or down, we have:

$$v_2 = \ln\frac{y_2}{x_2}$$

$$e^{v_2} = \frac{y_2}{x_2}$$

$$y_2 = x_2 e^{v_2}$$

Clearly, $y_1 = x_1$ and there is no rotation.

3.2. **3d Hyperbolic Geometry.**

3.2.1. *Finding the direction and distance from one point to another.* We will use the upper half space model: $\mathbb{H}^3 \mapsto \{(x_1, x_2, x_3) : x_3 > 0\}$.

Given points $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$, the first step is to find the vertical half-plane that intersects both points. This way, the problem can be reduced to a problem in $\mathbb{H}^2$. Unless both points are on the same vertical line i.e. $x_1 = y_1$ and $x_2 = y_2$, we let the horizontal distance between the two points be $u = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$, then set $\mathbf{w} = (w_1, w_2) = (y_1 - x_1, y_2 - x_2)/u$ to be the unit 2-vector on the horizontal plane in the direction from $\mathbf{x}$ to $\mathbf{y}$. We can now work with $(x_1', x_2') = (0, x_3)$ and $(y_1', y_2') = (u, y_3)$.

Once we have two points on a half-plane $\mathbf{x}', \mathbf{y}'$, we can use the two-dimensional solution to find the vector $\mathbf{v}' = (v_1', v_2')$ representing the direction and distance from $\mathbf{x}'$ to $\mathbf{y}'$.

We now translate the vector back from the half-plane using $\mathbf{v} = (v_1, v_2, v_3) = (v_1' w_1, v_1' w_2, v_2')$.

The distance remains the same as it was in the two-dimensional case.

There is a problem if $x_1 = y_1, x_2 = y_2$ because $\mathbf{w}$ is undefined. In this case, $\mathbf{v} = (0, 0, \ln \frac{y_3}{x_3})$. Alternately, we could use any unit vector for $\mathbf{w}$ above, since the horizontal displacement is zero in any direction.

3.2.2. *Finding the point a given distance in a given direction from another.* Given initial point $\mathbf{x} = (x_1, x_2, x_3)$ and vector $\mathbf{v} = (v_1, v_2, v_3)$, we first reduce to the $\mathbb{H}^2$ case as before, unless $v_1 = v_2 = 0$. Let $u = \sqrt{v_1^2 + v_2^2}, \mathbf{w} = (w_1, w_2) = \frac{(v_1, v_2)}{u}$ and solve the two-dimensional case using the point $\mathbf{x}' = (x_1', x_2') = (0, x_3) \in \mathbb{H}^2$ and vector $\mathbf{v}' = (v_1', v_2') = (u, v_3) \in \mathbb{R}^2$ to obtain the point $\mathbf{y}' \in \mathbb{H}^2$ and the angle of rotation θ.

Now we just need to map $\mathbf{y}'$ back to $\mathbb{H}^3$. The first two coordinates are $(x_1, x_2) + y_1' \mathbf{w} = (x_1 + y_1' w_1, x_2 + y_1' w_2)$. The third coordinate is y_2'. This gives the full coordinates as $(x_1 + y_1' w_1, x_2 + y_1' w_2, y_2')$.

We will also need to find the change in orientation. Our solution in the two-dimensional analogue gave us the angle of rotation, θ. This corresponds to the point of reference being rotated with the rotation matrix $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$.

If $\mathbf{w}$ happens to be $(1, 0)$, then the $\mathbb{H}^2$ we're using is the subspace of $\mathbb{H}^3$ where the second coordinate is constant. In this case, we're just performing that rotation on the first and third coordinates, resulting in the matrix $M = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$.

In general, we can rotate the coordinate system so that $\mathbf{w} = (1, 0)$, perform the rotation M, and then rotate it back. This is equivalent to conjugating M

with $\begin{pmatrix} w_1 & w_2 & 0 \\ -w_2 & w_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. This gives

$$\begin{pmatrix} w_1 & -w_2 & 0 \\ w_2 & w_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} w_1 & w_2 & 0 \\ -w_2 & w_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

If $v_1 = v_2 = 0$, then $\mathbf{y} = (x_1, x_2, x_3 e^{v_3})$ and there is no rotation.

3.3. **2-Wormhole.** Consider the quotient space of $\mathbb{H}^2$ generated by quotienting out by a loxodromic transformation. We can set the half-plane model $\mathbb{H}^2 = \{(x, y) : y > 0\}$ so that the axis of the loxodromic transformation is the $x = 0$ geodesic. The loxodromic transformation thus becomes $(x, y) \mapsto e^k(x, y)$ where k is the hyperbolic distance translated along the axis.

The above quotient identifies the semicircles in figure 4. The resulting quotient space is homeomorphic to $S^1 \times \mathbb{R}$.

The family of functions $(x, y) \mapsto e^d(x, y)$ maps to automorphisms on this quotient space. Since these automorphisms fix radial lines, it would be convenient for one coordinate in the quotient space to be along radial lines. In addition, semicircles about the origin correspond to geodesics, and are fixed by automorphisms in the family $(x, y) \mapsto e^d \left(\frac{1}{x}, \frac{1}{y}\right)$, which are the point inversions about the origin. It will be convenient for the other coordinate in the quotient space to move along these geodesics.

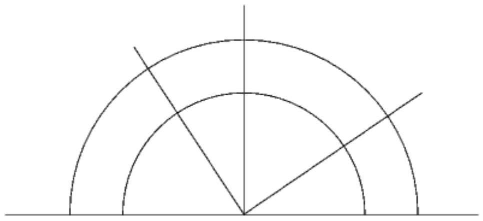


Figure 4. Coordinate Curves for 2-Wormhole

The obvious coordinate system to allow these coordinate curves is polar coordinates: $(r, \phi) = \left(\sqrt{x^2 + y^2}, \arctan \frac{y}{x}\right)$. This will be a useful starting point, but it can be further improved.

Under polar coordinates, multiplying x and y by e^k corresponds to multiplying r by e^k. This increases $\log r$ by k, and thus increases $\frac{2\pi}{k} \log r$ by 2π. If we use $\frac{2\pi}{k} \log r = \frac{2\pi}{k} \log \sqrt{x^2 + y^2} = \frac{\pi}{k} \log(x^2 + y^2)$ as the first coordinate, then the quotient on $\mathbb{H}^2$ is equivalent to quotienting the coordinate by addition of 2π. Since taking that quotient on $\mathbb{R}$ results in S^1, we can let $\theta = \frac{\pi}{k} \log(x^2 + y^2)$ be one of the coordinates.

I have found that rather than using θ as a coordinate in $\mathbb{R}/[2\pi]$, it's simpler to use θk as a coordinate in $\mathbb{R}/[2\pi k]$. Since $\theta k = \pi \log(x^2+y^2)$, the computations will be independent of k, and I just change k by changing the quotient on $\mathbb{R}$.

The choice for the other coordinate is less significant. Any monotonic function of ϕ will work. We will use $t = \cot\phi = \frac{x}{y}$, since it's easy to calculate. I chose $\frac{x}{y}$ rather than $\frac{y}{x}$ because $y > 0$, so $\frac{x}{y}$ is always defined.

This gives the coordinate system

$$(t, \theta k) = \left(\frac{x}{y}, \pi \log(x^2+y^2)\right)$$

for the 2-Wormhole.

In addition to mapping the points between the half-plane coordinates and the 2-Wormhole coordinates, we have to be able to convert vectors between the two. The vectors used in the half plane model are set so that $(0,1)$ points to the point at $(0,\infty)$ and $(1,0)$ is 90° clockwise of this.

For the 2-Wormhole, at (x,y) in half-plane coordinates, the 2-wormhole vector $(0,1)$ should point in the direction of $(1+\epsilon)(x,y)$, and $(1,0)$ should point 90° counter-clockwise of this. We'll now compute the matrices to transform between these vector coordinates.

Suppose you're at point (x,y) in $\mathbb{H}^2$. Let $\phi = \cot^{-1}\frac{x}{y} = \tan^{-1}\frac{y}{x}$. The vector $(0,1)$ in wormhole coordinates should correspond to the direction of (x,y) in $\mathbb{H}^2$ coordinates. This is $(\cos\phi, \sin\phi)$.

The vector $(1,0)$ in wormhole coordinates should correspond to the direction 90° counter-clockwise. This comes out to $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix} = \begin{pmatrix} -\sin\phi \\ \cos\phi \end{pmatrix}$.

Using this, the matrix to convert from wormhole coordinates to $\mathbb{H}^2$ coordinates should be $\begin{pmatrix} -\sin\phi & \cos\phi \\ \cos\phi & \sin\phi \end{pmatrix}$. This matrix is its own inverse, so it can be used to convert either way.

The final problem is finding the rotation of a point that has moved from p to q. Since this is an orientation-reversing map, the rotation must be inverted.

The angle of counter-clockwise rotation ψ' on the wormhole map is $-\psi - (\phi_1 - \phi_0)$ where ψ is the angle of counter-clockwise rotation from the parallel transport along the geodesic between p and q on the upper half-plane, ϕ_0 is $\cot^{-1}\frac{x_0}{y_0}$ where (x_0, y_0) are the half-plane coordinates of p, and ϕ_1 is $\cot^{-1}\frac{x_1}{y_1}$ where (x_1, y_1) are the half-plane coordinates of q.

This gives the rotation matrix $\begin{pmatrix} \cos\psi' & -\sin\psi' \\ \sin\psi' & \cos\psi' \end{pmatrix}$.

3.4. Surface Of Revolution.

In a 2-dimensional surface of revolution, a curve is rotated around an axis and each point traces out a circle. This results in a surface that is preserved under rotation. That is to say, it has the symmetries of a circle. A surface of revolution can easily be generalized to higher dimensions by rotating a curve around an axis so each point traces an n-sphere in $\mathbb{R}^{n+2}$.

My program relies largely on intrinsic geometry, so it would be helpful to modify this definition to use the intrinsic geometry of a manifold. Rather than looking at how the manifold is formed, I will look at the symmetries it inherits.

Let $G \leq \mathrm{Aut}(S^n \times \mathbb{R})$ be the subgroup of automorphisms on $S^n \times \mathbb{R}$ that fix the $\mathbb{R}$ coordinate.

A possible generalization of a n-surface of revolution as a manifold M such that there exists a homeomorphism $f : S^{n-1} \times \mathbb{R} \to M$ that preserves the automorphisms in G.

This definition introduces a few problems. For example, $\mathbb{S}^2$ would not be considered a surface of revolution, since there is no homeomorphism from $\mathbb{S}^1 \times \mathbb{R}$ to S^2. We can fix this by not requiring f to be injective.

Instead, I define an n-surface of revolution to be a manifold M such that there exists a continuous function $f : S^{n-1} \times \mathbb{R} \to M$ where given any automorphism g in G, $f(x) \mapsto f \circ g(x)$ is well-defined and is an automorphism.

In Section 1.3, we discussed working with Wormhole. This was a special case of a method for working with 3-surfaces of revolution.

We can reduce the problem of working on a 3-surface of revolution (or even an n-surface of revolution) to working on a corresponding 2-surface of revolution.

Consider a point x on a 3-manifold U that is a surface of revolution and a vector v in the tangent space U_x.

Consider the component v_0 of v along S^2.

Consider the great circle made by extending x to a geodesic in the direction of v_0.

We can reflect the spherical coordinate across this great circle. This is an automorphism on $S^2 \times \mathbb{R}$, and is therefore an automorphism on U.

By symmetry, the geodesic made by extending x in the v direction must stay on this slice of U.

This reduces finding the geodesic on a 3-surface of revolution to finding one on a 2-surface of revolution.

3.4.1. *Finding the vectors from a point:* Rather than mapping a sphere to $\mathbb{R}^2$, we can embed it in $\mathbb{R}^3$. Given points $\mathbf{x} = (\mathbf{x}_1, x_2), \mathbf{y} = (\mathbf{y}_1, y_2) \in S^2 \times \mathbb{R}$, let $\mathbf{v}$ be the unit vector in the direction of $\mathbf{y}_1$ from $\mathbf{x}_1$. You can find this by looking at $\mathbf{x}_1$ and $\mathbf{y}_1$ as vectors in $\mathbb{R}^3$, taking the projection of $\mathbf{y}_1$ perpendicular to $\mathbf{x}_1$, and normalizing.

Let θ be the angle between $\mathbf{x}_1$ and $\mathbf{y}_1$, so $\theta = \arccos \langle \mathbf{x}_1, \mathbf{y}_1 \rangle$.

Now we take the points $\mathbf{x}' = (0, x_4)$ and $\mathbf{y}' = (\theta, y_4)$ in $S^1 \times \mathbb{R}$.

Let $\mathbf{z}'$ be a vector between them.

Let the S^2 component of $\mathbf{z}$ be $z_0'\mathbf{v}$ and the $\mathbf{R}$ component be z_1'.

$\mathbf{z}$ is a vector between the two points.

3.4.2. *Finding a point from a vector:* Given point $\mathbf{x}$ and vector $\mathbf{z}$,

Let $\mathbf{x}' = (0, x_2), \mathbf{z}' = (\|\mathbf{z}_1\|, z_2)$.

Let $\mathbf{v} = \|\mathbf{z}_1\|$.

Find $\mathbf{y}'$ with the two-dimensional version.

$\mathbf{y} = (\mathbf{x}_1 \cos y_1' + \mathbf{v} \sin y_1', y_2')$.

In order to find the rotation, we must work in a more relevant basis. We can do this by commuting with the appropriate matrix.

$\mathbf{e}_1 \mapsto \mathbf{e}_1, \mathbf{v} \mapsto \mathbf{e}_2, \mathbf{x}_1 \mapsto \mathbf{e}_3, \mathbf{x}_1 \times \mathbf{v} \mapsto \mathbf{e}_4$

$(\mathbf{e}_1, \mathbf{v}, \mathbf{x}_1, (\mathbf{x}_1 \times \mathbf{v}))^{-1}$

$= (\mathbf{e}_1, \mathbf{v}, \mathbf{x}_1, (\mathbf{x}_1 \times \mathbf{v}))^T$, since it's a rotation matrix.

First, we use the rotation of the two-dimensional version to find the rotation between $\mathbf{e}_1$ and $\mathbf{v}$.

$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Next, we have to deal with the fact that the S^2 component rotates.

Let $\phi = y_2' - x_2'$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Putting this all together, we get:

$$(\mathbf{e}_1, \mathbf{z}, \mathbf{x}, (\mathbf{x}{\times}\mathbf{z})) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (\mathbf{e}_1, \mathbf{z}, \mathbf{x}, (\mathbf{x}{\times}\mathbf{z}))^T$$

3.4.3. *Portals:* The intersection tells me the orientation as embedded in $\mathbb{R}^3$. I need to convert that to $\mathbb{R} \times S^2 \subseteq \mathbb{R}^4$.

First, I just expand the matrix M to $\begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}$ to put it in $\mathbb{R}^3 \subseteq \mathbb{R}^4$.

Now I need to reflect it twice.

First, I will reflect along the t-axis, to get $\begin{pmatrix} M & 0 \\ 0 & -1 \end{pmatrix}$. This doesn't change anything, since that vector was orthogonal to reality, but it does guarantee that the final orientation will have the same sign.

Now I need to reflect between $(0, 0, 0, 1)$ and $(v, 0)$ where $v \in S^2$ is the position of the vector.

Let w be a vector I am moving with this, and R be the reflection matrix.

$Rw = w - ((v,0) - (0,0,0,1)) \langle (v,0) - (0,0,0,1), w \rangle$
$= w - (v,-1) \langle (v,-1), w \rangle$
$= Iw - (v,-1)(v,-1)^T w$
$= (I - (v,-1)(v,-1)^T)w$
$R = I - (v,-1)(v,-1)^T$

This means that the final orientation is $R \begin{pmatrix} M & 0 \\ 0 & -1 \end{pmatrix}$.

Or rather, it would be, except that the real coordinate is the first coordinate, not the last. I just need to cycle it through with the matrix

$$S = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Giving me

$$SR \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}$$

References

1. John Lott Bruce Kleiner, *Locally collapsed 3-manifolds*, http://arxiv.org/abs/1005.5106, May 2010.
2. Charlie Gunn, *Computer graphics and mathematics*, ch. Visualizing hyperbolic geometry, pp. 299–313, Springer, 1992.
3. Charlie Gunn and Delle Maxwell, *Not knot*, 1991.
4. William Jaco and Peter B. Shalen, *A new decomposition theorem for irreducible sufficiently-large* 3-*manifolds*, Algebraic and geometric topology (Proc. Sympos. Pure Math., Stanford Univ., Stanford, Calif., 1976), Part 2, Proc. Sympos. Pure Math., XXXII, Amer. Math. Soc., Providence, R.I., 1978, pp. 71–84. MR 520524 (80j:57008)
5. ———, *Seifert fibered spaces in* 3-*manifolds*, Geometric topology (Proc. Georgia Topology Conf., Athens, Ga., 1977), Academic Press, New York, 1979, pp. 91–99. MR 537728 (80k:57016)
6. William H. Jaco and Peter B. Shalen, *Seifert fibered spaces in* 3-*manifolds*, Mem. Amer. Math. Soc. **21** (1979), no. 220, viii+192. MR 539411 (81c:57010)
7. Jian Ge Jianguo Cao, *Perelman's collapsing theorem for 3-manifolds*, http://arxiv.org/abs/0908.3229, Aug 2009.
8. Klaus Johannson, *Homotopy equivalences of* 3-*manifolds with boundaries*, Lecture Notes in Mathematics, vol. 761, Springer, Berlin, 1979. MR 551744 (82c:57005)
9. Rune Johansen, *Through the wormhole portal*, http://runevision.com/3d/anims/#52, August 2003.
10. Gang Tian John Morgan, *Completion of the proof of the geometrization conjecture*, http://arxiv.org/abs/0809.4040, September 2008.
11. H. Kneser, *Geschlossen flächen in dreidimensionalen mannigfaltigkeiten*, Jahresbericht der Deutschen Mathematiker Vereinigung **38** (1929), 248–260.
12. J. Milnor, *A unique decomposition theorem for 3-manifolds*, American Journal of Mathematics **84** (1962), no. 1, 1–7.

13. Michael S. Morris and Kip S. Thorne, *Wormholes in spacetime and their use for interstellar travel: a tool for teaching general relativity*, Amer. J. Phys. **56** (1988), no. 5, 395–412. MR 941172 (89d:83001)
14. Grisha Perelman, *The entropy formula for the ricci flow and its geometric applications*, http://arxiv.org/abs/math.DG/0211159, November 2002.
15. ______, *Finite extinction time for the solutions to the ricci flow on certain three-manifolds*, http://arxiv.org/abs/math.DG/0307245, July 2003.
16. ______, *Ricci flow with surgery on three-manifolds*, http://arxiv.org/abs/math.DG/0303109, March 2003.
17. Mark Phillips and Charlie Gunn, *Visualizing hyperbolic space: Unusual uses of 4x4 matrices*, The National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center) (1991), 209–214.
18. Tamara Munzner Stuart Levy and Mark Phillips, *Geomview*, http://www.geomview.org/, August 2007.
19. Jeff Weeks, *Curved spaces*, http://www.geometrygames.org/CurvedSpaces/index.html.
20. ______, *Real-time rendering in curved spaces*, Computer Graphics and Applications **22** (2012), no. 6, 90–99.
21. Corvin Zahn, *Flight through a wormhole*, http://www.spacetimetravel.org/wurmlochflug/wurmlochflug.html, March 2008.