

I am writing a graphics program for three-dimensional non-euclidean geometry. I currently have  $\mathbb{E}^3$  and  $\mathbb{H}^3$  implemented.

I hope to implement a method to glue multiple spaces together along spherical boundaries.

Outside of Euclidean geometry, gluing spaces together like this won't always be smooth. For example, trying to glue a spheres from Euclidean and hyperbolic geometry. 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 you glue the insides or outsides of two spheres together.

In order to get around this problem, I have designed and am currently in the process of implementing what I call wormhole.

I define an  $(n, m)$ -surface of revolution as an  $(n + m)$ -surface with an automorphism group that has the automorphism group of the  $n$ -sphere as a subgroup. Note that an  $(n, m)$ -surface of revolution is also an  $(n - 1, m + 1)$ -surface of revolution for all  $n \geq 1$ . For example,  $S^2$  is a  $(2, 0)$ -surface of revolution, so it is also a  $(1, 1)$ - and  $(0, 2)$ -surface of revolution. In addition, this definition is defined entirely by the intrinsic geometry of the space, and does not require embedding into Euclidean geometry.

I define a 2-wormhole as a quotient space on  $\mathbb{H}^2$  made by identifying two ultraparallel lines. Note that this is a  $(1, 1)$ -surface of revolution.

An  $n$ -wormhole is an  $(n - 1, 1)$ -surface of revolution. This does not generally have constant curvature, and is not a quotient space on  $\mathbb{H}^n$ . However, I have found a method to reduce finding a geodesic on an  $(n, m)$ -surface of revolution to a  $(1, m)$ -surface of revolution. In particular, I can find the geodesic on an  $n$ -wormhole using the method to find a geodesic on a 2-wormhole.

There are two other spaces that are similar to this wormhole that are of interest. I call one a black hole and the other a cone point space. These spaces, and the wormhole, are all built by extending to the third dimension a quotient space of  $\mathbb{H}^2$  in which two lines are identified. The wormhole occurs if the two lines are ultraparallel, the black hole if they're asymptotic, and the cone point space if they intersect.

Black holes and cone spaces can allow you to connect two spaces in a similar manner, but they're more interesting just connected to one. An  $n$ -black hole is an  $n$ -dimensional analogue of a pseudosphere, in which the embedding into  $\mathbb{E}^3$  which has spherical cross-sections instead of circular ones. An  $n$ -cone point space is notable for containing a cone point and still being easy to connect to  $\mathbb{E}^n$  or  $\mathbb{H}^n$ .

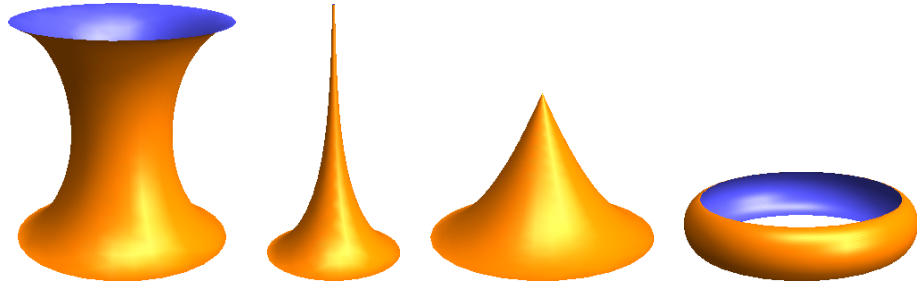
Another way to generalize this is to identify lines on  $\mathbf{E}^2$  or  $\mathbf{S}^2$  instead of  $\mathbf{H}^2$ .

Identifying two parallel lines on  $\mathbf{E}^2$  creates a cylinder, which extends to  $\mathbb{S}^n \times \mathbb{R}$ . Identifying two intersecting lines creates a cone, which extends to a higher dimensional cone. Identifying two lines on a sphere creates something shaped like a lemon, with two cone points on opposite sides. If you extend this to identifying two lines that meet at an angle greater than  $360^\circ$  ???

There are three more surfaces of revolution of constant curvature. One is like a sphere, but with two cone points on opposite sides. If part of it is embedded into  $\mathbb{E}^3$ , it resembles the outside of a torus. I will call this the extended sphere. The other two are a cylinder and a disk, which extend to  $\mathbb{E} \times \mathbb{R}^2$  and  $\mathbb{E}^3$ . These two surfaces do not require this technique to implement.

The extended sphere is interesting for the parts of it that cannot be embedded in  $\mathbb{E}^3$ . Gluing one of these to  $\mathbb{E}^3$  will result in a cone point with more than 360 degrees, which cannot be done in the cone point space. Furthermore, a piece of  $\mathbb{H}^3$  can be glued into the middle, replacing the cone point, to create a smooth space that's smaller on the inside.

Figures 1-4 illustrate the 2-dimensional versions of these surfaces. Note that while only a portion of the surfaces can be embedded into  $\mathbb{E}^3$ , my program will be capable of using the entire surface.



(a) Part of a 2-wormhole (b) Part of a 2-black hole point space (c) Part of a 2-cone (d) Part of a 2-extended sphere

I also hope to implement  $S^3$  and  $S^2 \times \mathbb{E}$ . I may also implement  $\mathbb{H}^2 \times \mathbb{E}$ , but it doesn't combine well with the wormhole, so it won't be as useful.

One difficulty in this project will be finding geodesics between two given points when spaces are glued together. In the spaces I have chosen, it is not difficult to construct a geodesic between two given points. However, this does not apply to the compound spaces made by gluing them together.

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 distance they are from the target point, I can use Newton's method to numerically solve for the geodesic between two given points.

Alternately, I can build a raytracer. Rather than finding the direction each point is in from the camera, I construct geodesics corresponding to each pixel in the camera, and check to see what's in that direction. This runs into problems with light sources, since I still need to find the angle and distance of the light source from the given point. I have found three possible methods to get around this:

I can give every point the same lighting, regardless of position. This will likely be useful for debugging, but it results in no shading, which will make it less useful or interesting.

I can use the camera as the light source. In this case, the geodesic that passes between the camera and a given point is the one I used to find that point to begin with. The shading will only be a function of distance, unless phong shading is used. It will look better, but it will still be very limited.

I can simply fire photons in various directions from the light sources, and use them to color objects they strike. Photons can be used to give effects that are otherwise impossible, but they generally take a very long time to calculate.

There is another notable detail. In euclidean geometry, lighting falls off inversely with the square of distance. This does not apply in general. Changing the direction and distance a photon is emitted by  $\varepsilon \mathbf{u}$  for some unit vector  $\mathbf{u}$  will change where it hits by  $M\varepsilon \mathbf{u} + O(\varepsilon^2)$  for some matrix  $M$ . The brightness is inversely proportional to  $|\det(M)|$ .

More accurately, given the list of all vectors  $\mathbf{v}_1, \mathbf{v}_2, \dots$  such that the photons moving in that direction for that distance hit the given point, the brightness at that point is proportional to  $\sum \frac{1}{|\det(M_{\mathbf{v}_i})|}$ . This is because, in general, two points can be connected by more than one geodesic.