Virtual Crocheting of Euclidean Planes in a 3-Sphere

Eryk Kopczyński[1] and Dorota Celińska[2]

[1]Institute of Informatics, University of Warsaw, Poland; erykk@mimuw.edu.pl
[2]Faculty of Economic Sciences, University of Warsaw, Poland; dcelinska@wne.uw.edu.pl

Abstract

Embeddings in Euclidean space make a popular way of visualizing non-Euclidean geometries. A typical example are embeddings of fragments of hyperbolic space in the three-dimensional Euclidean space using hyperbolic crocheting and similar methods. In this article we discuss the opposite problem: how to embed a Euclidean plane in a three-dimensional spherical space.

Introduction

The fifth axiom of Euclidean geometry was perceived as a problematic one, because it raised some questions about the nature of parallelness, which were unanswered up to the 19th century. Take a line L and a point A. According to Euclidean geometry principles, there is exactly one line going through A which does not cross L. But can there be more? Or less? The answers to those questions can be found in non-Euclidean geometries. The first, and probably the most famous one, is hyperbolic geometry, discovered by Gauss, Lobachevsky and Bolyai. In this case, there are infinitely many lines going through A which do not cross L. While hyperbolic geometry is not common in our world (typical examples include coral reef or lettuce), the second kind of non-Euclidean geometry is more common—that is, the geometry of the sphere.
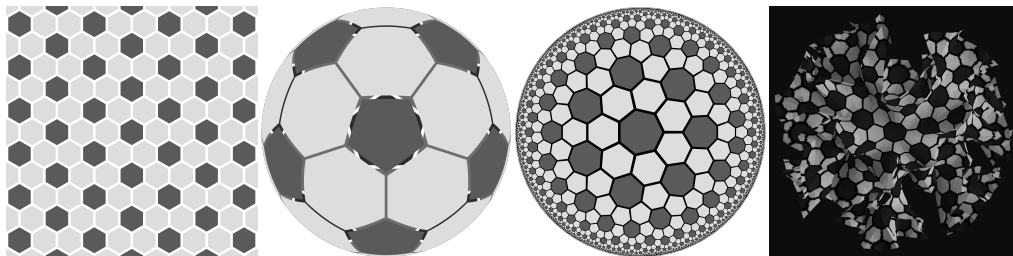
Figure 1: *Hexagonal grids and curvature. From left to right: Euclidean hex grid, soccerball, hyperbolic soccerball in the Poincaré model, tangible hyperbolic soccerball (simulated).*

The surface of a sphere is an isometric 3D model of spherical geometry. Can we get a similar physical model of hyperbolic geometry? One of the possible methods is hyperbolic crocheting [2]. Typical crocheting yields a plane; to crochet a hyperbolic plane, in each row (after the first, foundation row) we add an extra stitch every k stitches. The resulting model appears curvier with every added row. Another approach is to consider the hexagonal tiling of the Euclidean plane, as shown on Figure 1a. This tiling constructs a plane out of hexagons. When we replace the black hexagons with pentagons, we get a classic soccer ball, which is a good approximation of a sphere. A model of (a part of) the hyperbolic plane is obtained by replacing the black hexagons with heptagons. This process can be done with many materials, such as paper or beads (Figure 2).

As we can see, popular methods boil down to taking a construction of a plane and adding extra area to it. The surface area of a sphere is finite, and the surface area of a plane is infinite. The hyperbolic plane is,
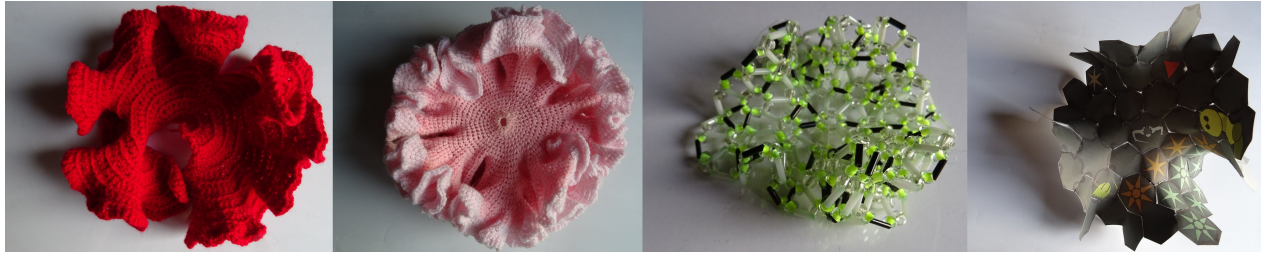
Figure 2: *Tangible models of hyperbolic plane. Left to right: two crochets, beaded, and paper model.*

roughly speaking, even more infinite than the Euclidean plane. In a Euclidean plane, the amount of area in radius r is on the order of r^2. In the hyperbolic geometry, for big enough r, the amount of space in radius r grows exponentially with r. This exponential growth aspect has found applications in data visualization [4] and also in the design of our game HyperRogue [3], which we recommend to gain intuitions about the geometry of the hyperbolic plane.

Because of the exponential growth, we can only crochet a small fragment of a hyperbolic plane. We live in a three-dimensional space, so the amount of space in radius r is on the order of r^3; this allows us to easily fit small fragments of hyperbolic plane, but our crochet, made of a material of positive uniform thickness, will fill more and more of the available space, and at some point, there will be not enough space to crochet further.

In this paper, rather than manufacturing a non-Euclidean surface within our Euclidean space, we try to go the other direction. Can we crochet a Euclidean plane in the three-dimensional spherical space, S^3? In many ways, the step from Euclidean and hyperbolic geometry is similar to the step from spherical to Euclidean. Therefore, the result should be similar to the usual hyperbolic crocheting. The limitations of our senses make solving the problem with tangible results quite a challenge. However, the technology allows us to use computer simulations to picture being inside a non-Euclidean space.

For us, Euclidean plane is a boring, perfectly natural thing, while a hyperbolic plane is strange, unnaturally big, too big to fit in our world. For creatures living in S^3, our common Euclidean plane will be as unnatural as the hyperbolic plane is unnatural for us. Our simulation allows to see the world from their perspective. Another major motivation of this work is that the perspective in S^3 seems to have so far received less attention than its hyperbolic [1] and periodic Euclidean counterparts. We believe that it is actually more interesting than these spaces, because of the phenomena that occur only in spherical perspective.

Perspective in the 3-Sphere

Suppose our eye is at point E in a three-dimensional space (Figure 3a). While traditional perspective works by taking a plane P and projecting every point A to the intersection of EA and P, this approach is not appropriate for non-Euclidean geometry, because planes are not well-defined there. Instead, it is more reasonable to take a sphere S centered at E. Again, we project every point A to a point A' which is the intersection of S and the ray EA. Thus, the perceived size of a segment AB is the spherical distance between A' and B', which is proportional to the angle AEB. The sphere S can be then projected back to a computer screen, which allows the viewer to see the world just as they would see it if they lived in the non-Euclidean world.

To understand the perspective in the 3-sphere, we will analyze the view of a flatlander living on a two-dimensional sphere. Our flatlander is only aware of the sphere itself, not of the three-dimensional world it is embedded in. She can only look in directions which are tangent to the sphere, and she cannot see anything which is not in the sphere, because the light rays in this world also stick to the sphere. Assume her eye E
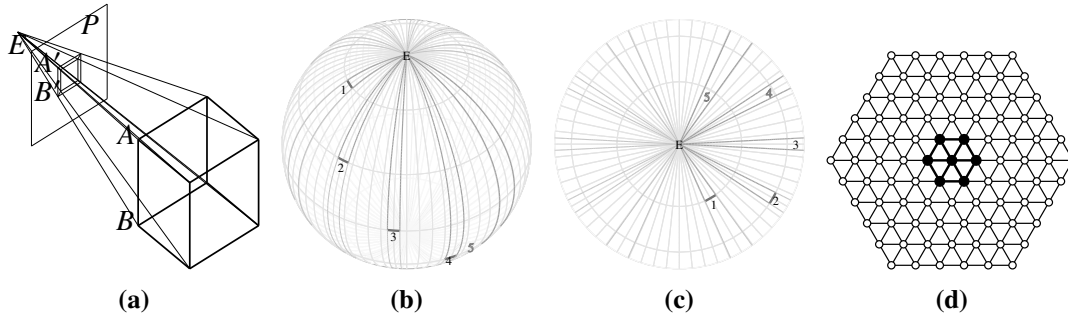
Figure 3: *(a) Classic Euclidean perspective. (b) Light rays on S^2 viewed from the side. (c) Light rays on S^2 viewed from above. (d) Hexagonal grid used as the base of virtual crocheting.*

is at the north pole, and thus, the light rays travel along meridians. What will be the perceived size of a segment AB? Figure 3bc shows five segments 1, 2, 3, 4, 5 of the same length on a sphere. Let A_i and B_i be the endpoints of segment i. The segment 2 is further away from E than the segment 1, and its perceived size (proportional to the angle A_2EB_2) is smaller than the perceived size of segment 1 (proportional to the angle A_1EB_1). While in the Euclidean geometry, this perceived size gets smaller and smaller as AB gets further from us, this is not the case on the sphere. The segment 3 appears the smallest; after crossing the equator, our segment starts to appear larger and larger – segment 4 appears as large as segment 2, and segment 5 appears as large as segment 1, and in particular, objects close to the south pole appear as large as if they were close to E! Another striking feature of the spherical perspective is that all the rays starting from our eye will travel along their great circles and eventually hit the back of our head. As a consequence of this, if we look the other direction, we see exactly the same scene, unless the objects have been obscured by something else. In particular, when we move past an object, or through an object, it will appear to move in front of us! These phenomena are evident in our simulation.

Virtual Crocheting Algorithm

In this section we briefly describe our algorithm which virtually places a fragment of the Euclidean plane in S^3. Our algorithm simulates building a surface from rigid beams. Consider the Euclidean hex grid shown in Figure 3d. This grid can be viewed as a graph $G = (V, E)$. We need to put our grid into S^3, or, in other words, for each $v \in V$ we need to find its location $v' \in S^3$; once we find the location of all the grid points, the location of all the non-grid points in our fragment are assumed to be given by interpolation. For each edge $e = (v, w)$, let $d_1(e)$ be the Euclidean distance between these vertices in G (this is constant in the hex grid, but may be not constant in general), and let $d_2(e)$ be distance between v' and w'. We want to find a mapping which is isometric, i.e., $d_1(e) = d_2(e)$. Such a mapping can be found with a process which is again a simulation of the physical rigid beams.

We start with the central vertex O and its six adjacent vertices A, B, C, D, E, F (marked with black circles in Figure 3d). In the Euclidean world, every angle of an equilateral triangle has $60°$, so these seven vertices can be easily arranged in the plane. However, trying to do the same in S^2 or S^3 will not work, since an equilateral spherical triangle has angles greater than $60°$. Therefore, we need to move A', B', C', D', E', F' both ways along the third dimension to make all the angles at the central vertex O' greater than $60°$. Our algorithm finds the arrangement in the following way: we start with a somewhat random mapping $v \mapsto v'$, and then, in each iteration, for every edge (v, w), we compute and compare $d_2(e)$ and $d_1(e)$. If $d_2(e) > d_1(e)$ we bring the two points closer, and if $d_2(e) < d_1(e)$, we bring them further apart. The simulation should converge to a mapping for which $d_2(e) = d_1(e)$. We iterate until $|d_2(e) - d_1(e)| < \epsilon$ for all edges.

After successfully mapping the seven central vertices, we gradually add the remaining points. We find

which points in our full grid are the closest to A, and add them to our construction; we use the embedding found so far to place every new point close to its correct position right from the start. We perform iterations as above until $|d_2(e) - d_1(e)| < \epsilon$ for all edges. To prevent the model from crossing itself, we also check for pairs of points (v, w) such that v' and w' are very close even though v and w are not, and if such pairs are found, we add a repelling force between them.

After the whole fragment is embedded, we subdivide each triangle into four smaller triangles by placing a new vertex in the midpoint of each edge, and iterate the process until $|d_2(e) - d_1(e)| < \epsilon$ for each edge again. The more vertices we have, the more accurate the model is, but also the slower it is to compute and to render. We repeat the process of subdivision until we achieve the desired number of vertices, and then, we simply let the model converge by reducing ϵ.

Our Simulation and Conclusions

We paint our fragment of the Euclidean plane with Euclidean versions of tesselations and characters from HyperRogue [3]. This not only brings life to the picture, but also provides useful landmarks to study perspective phenomena. The simulation can be accessed at http://roguetemple.com/z/virtual-crocheting/, in the following forms:

- An interactive browser-based simulation of Euclidean crocheting in S^3.

- A VR video which can be viewed with VR hardware. Cheap options include red-cyan anaglyph glasses and mobile-based VR sets such as the Google Cardboard. The viewer can choose the direction they are looking, but they cannot move the camera.

- An Android app which allows us to explore the simulation interactively on a mobile-based VR set.

Non-Euclidean crochets and similar models turn out to be very hard to appreciate on a two-dimensional picture, as it is very difficult to convey their three-dimensional nature. Virtual reality view, or even simply letting the user to interactively rotate the simulation, helps a lot here, but non-Euclidean crochets need to be touched to be fully appreciated. This is because they are dynamic objects – even though their creases appear to be random, their curvature is constant, and any fragment can be folded to exactly fit the curvature of any other fragment. Our simulation allows us to alter the folding, but in a very restricted way. Further advances in VR technology should allow us to play with the virtual crochets in our hands.

The embedding algorithm is a generalization of the Hypersian Rug algoritm in HyperRogue, which is itself inspired by the video *Growing a Hyperbolic Plane* by Riccardo Antonelli. Thanks to Christopher King for the idea of creating a VR video.

References

[1] V. Hart, A. Hawksley, E. A. Matsumoto, and H. Segerman. "Non-euclidean Virtual Reality I: Explorations of $\mathbb{H}^3$." *Bridges Conference Proceedings*, Waterloo, Ontario, Canada, July 27–31, 2017, pp. 33–40. http://archive.bridgesmathart.org/2016/bridges2017-33.html.

[2] D. W. Henderson and D. Tamina. "Crocheting the Hyperbolic Plane." *Mathematical Intelligencer*, vol. 23, no. 2, 2001, pp. 17–28.

[3] E. Kopczyński, D. Celińska, and M. Čtrnáct. "HyperRogue: Playing with Hyperbolic Geometry." *Bridges Conference Proceedings*, Waterloo, Ontario, Canada, July 27–31, 2017, pp. 9–16. http://archive. bridgesmathart.org/2016/bridges2017-9.html.

[4] T. Munzner. "Exploring Large Graphs in 3d Hyperbolic Space." *IEEE Computer Graphics and Applications*, vol. 18, no. 4, 1998, pp. 18–23.