Drawing on Surfaces in VR

Alexander Adrahtas David Dumas Alexander Guo Gregory Schamberger
University of Illinois at Chicago

Mathematical Computing Laboratory

UIC

Summary

This project is an extension of the development of the GenusView drawing tool by David Dumas. GenusView shows the relationship of a genus two surface and a corresponding tiling of the hyperbolic plane (denoted $\mathbb{H}^2$). The main function of the program is to allow the user to draw on either the genus two surface or on $\mathbb{H}^2$ and see the corresponding drawing appear on the other object.

The goal of this project was to create a virtual reality version of this drawing tool, which we call GenusLab. This allows for hand controlled movements and a comfortable, accurate view. We also added new functionality, such as a hole drilling feature that allows you to inspect inside of the genus 2 surface by removing parts of the tiling, a color wheel to select a drawing color, and a user interface controlled in ways that are natural for virtual reality. This project was conducted in the MCL in Fall 2018.

Motivation

Closed surfaces can be constructed by gluing sides of a polygon. Associated to such a gluing description there is a tiling of the Euclidean or hyperbolic plane. These tilings can be used to study the unique geometric qualities and differences of Euclidean and hyperbolic space. While this topic has been well studied and understood, there is a lack of tools for creating accurate models and pictures of this relationship. With a tool like GenusLab, we can more easily see the correspondence between a surface and $\mathbb{H}^2$.

Materials

To create our application we used the game engine Unity, which supports the Oculus VR system. For more realistic interactivity, our application utilizes the Oculus Touch hand controllers. Unity provides a 3D environment that allows us to create objects through C# code or in the scene editor. Every C# script is attached to one or more objects to handle interaction between these objects at run time. The scene editor is used to position objects and attach components such as scripts, physics systems, etc.. The tilings of the genus two surface and of $\mathbb{H}^2$ are created using shaders, which are 3D graphics programs that run on the GPU. The shader language that Unity uses is ShaderLab, which is based on the shader languages Cg and HLSL, which are in turn based on the C programming language.

Hyperbolic Geometry

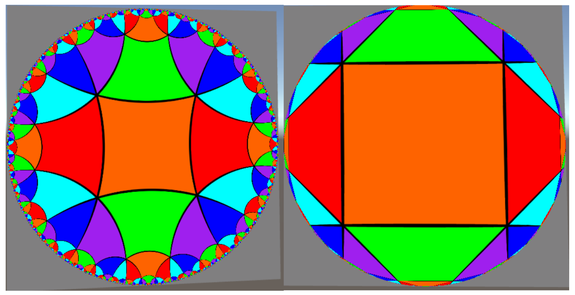


Figure: Poincaré (left) and Klein (right) models of the hyperbolic plane

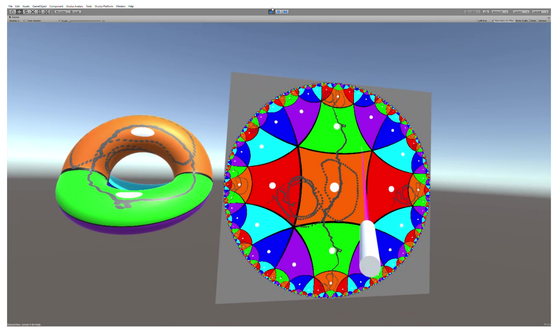
There are two models of $\mathbb{H}^2$ that the program lets the user toggle between. The tiles that our program works with are quadrilaterals with geodesic edges (as can be seen in the Klein model, which preserves straightness of geodesics). At each vertex, six identical tiles meet, and therefore each quadrilateral has internal angles of 60° (as can be seen in the Poincaré model, which preserves angles).

Added Features

In addition to bringing GenusView to VR, we created the following new features:

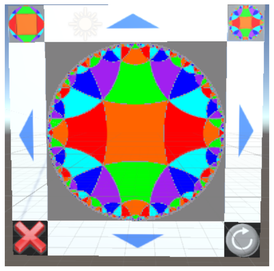
- Hole drilling tool
  Allows the user to examine the inside of the genus two surface by drilling a hole through the object at any point.
- Sliding hyperbolic view
  Allows the user to shift the perspective by sliding $\mathbb{H}^2$ plane up, down, left, or right.
- Physical controls to toggle between Poincaré and Klein models
- Toggling between color schemes
  We now support six different ways of coloring the tiling and the genus two surface, to emphasize different features. These include options to draw on a blank or fully transparent background.
- Color Wheel to control drawing color

Laser Pointer



When thinking of a drawing utensil in VR, the first thought is to create something familiar such as a marker. However, trying to draw on a surface that doesn't physically exist creates a disconnect between what the user sees and what the user feels. An alternative to this problem is the implementation of a laser pointer. The laser pointer allows a precise drawing experience that more closely simulates its real life counterpart.

User Interface



GenusView was controlled with keyboard and mouse. To make GenusLab a more comfortable VR experience we replaced these with buttons in the scene controlled through touch. Each button contains the ButtonDriver C# script that performs an action when another object enters it. Unity has built-in support for detecting collisions between objects through its "RigidBody" physics component. This makes such virtual physical controls easy to implement. We attach a RigidBody to the laser and hands, thus activating the buttons on collision. The ButtonDriver script then activates a function called OnTriggerEnter. This function sends a message to the Event Manager that calls the function in another script to handle the request. The RigidBody of the laser pointer is only active when the trigger on the back of the controller is held down.

Drawing

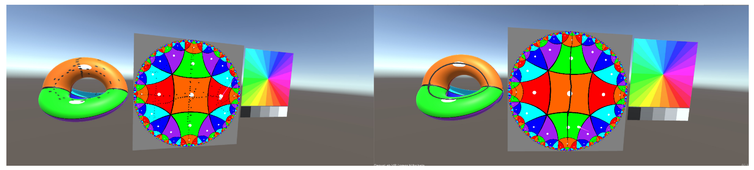
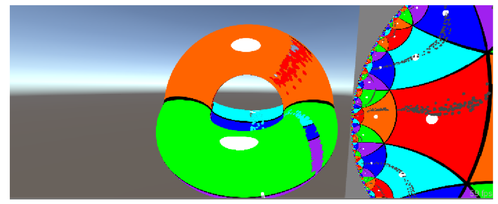


Figure: Original drawing (left), drawing with linear interpolation (right)

In GenusView, a point on the surface is drawn only once per frame. When moving the hand controller quickly, this results in a dotted line. To fix this, we added a function that draws additional points using linear interpolation. For each frame, we take the current detected point and the previous detected point and draw a number of points between them. We also added the ability to change the drawing color. When the laser is pointed at the color chooser, we determine the (u,v) coordinates on the texture that the laser is pointing at. Unity then uses the coordinates to determine which color is found at that point on the texture.

Hole Drilling Tool



The hole drilling tool creates a hole at any point of the surface or $\mathbb{H}^2$. Internally, this is accomplished by making that part of the surface fully transparent. This allows the user to inspect the inside of the surface. We created a shader that ensures that the faces of the genus two surface are drawn in the proper order. This is necessary so that the back faces can be seen through transparent parts of the front faces. The bright coloring of the genus two surface and the lack of shading and shadows resulted from disabling Unity's built-in standard shader, which was necessary for our transparency system to work properly.

Challenges

Unity is a very large and powerful engine; as such, it was common to get confusing errors that novice users such as ourselves had a difficult time overcoming. We often struggled getting our planned features working. A deeper understanding of topics such as transparency with shaders, behaviour and properties of built-in Unity components, and VR best practices were required. Any future development would probably require further understanding of these topics. VR also requires a much higher frame rate than traditional game development, limiting what calculations are possible for each frame.

Future Additions

The original keyboard and mouse control system of GenusView has been removed. Restoring the keyboard and mouse functionality (and making it work with our newly added features) would be our next step. This will allow us to broaden our audience from strictly VR users to anyone who owns a computer. Further possible features include allowing the genus two surface to be cut into pieces, changing the shape of the surface, preserving user drawings when changing the background texture, drawing a solid line at any drawing speed, drawing with a marker, and a more interactive VR experience with a more aesthetically pleasing scene.