

For my project, I am going to implement the stripe pattern algorithm described in <https://www.cs.cmu.edu/~kmcrane/Projects/StripePatterns/paper.pdf> The algorithm described by the paper uses the ideas of curvature on meshes and vector fields to calculate appropriate stripe patterns for the surface. These patterns are similar to those seen in nature on zebras or fingerprints. Computing these patterns allows for a procedural method for generating the stripes that would otherwise need to be done by hand. There are many existing algorithms for computing these types of patterns; however, the algorithm described by Knoppel et. al. in the mentioned paper is one of the most recent and state-of-the-art. The paper also mentions possible applications of the algorithm to non-photorealistic rendering. I would like to explore some of these applications, but in particular, I plan on trying to use these stripes to render brush strokes on the surface. My idea is to use the stripes output by the algorithm and leverage the colors of the input mesh to create artificial paint brush strokes on the surface of the object. I could also corrupt the stripes a little bit by changing some of the algorithm's parameters to yield much less uniform stripes.

I will likely be doing the implementation in C++, but may choose to use Python should the computation be too messy for C++ (Python libraries could make them much easier) and if computation time is not a huge concern.