Contoured Watercolor Rendering of 3D Meshes

Capstone Project for Vivian Morgowicz (2016)

Group Members: Vivian Morgowicz , Divya Mahadevan, Josh Tveite, and Michelle Lin

Abstract

The goal of our project was to implement the paper "Computing smooth surface contours with accurate topology" by Pierre Benard, Aaron Herzmann, and Michael Kass with a non-photorealistic rendering technique. For the latter, we decided to implement "Interactive watercolor rendering with temporal coherence and abstraction" by Adrien Bousseau, Matt Kaplan, Joelle Thollot and Francois X. Sillion.

We implemented the project mainly in C++ and OpenGL. To do the remeshing, we began by loading in models with the help of the ASSIMP library, then using the OpenSubdiv library to turn a rough mesh into a smoother Catmull-Clark subdivision surface. Next, with the information from the latter and a point of view, we modified the vertex information of the mesh in order to insert more triangles where the contours of the mesh would fall from that point of view. We also built a powerful debugger with a variety of options in order to ensure that we implemented this part of the process correctly.

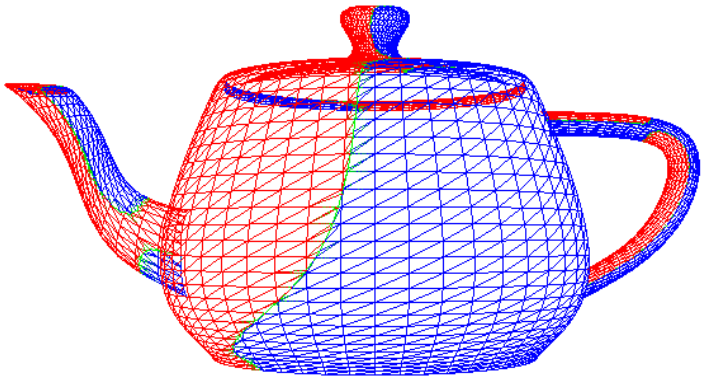
After remeshing, we needed to compute and render the contours of the mesh. Our original plan was to follow the paper and modify the Freestyle library, but the results were underwhelming. Freestyle was fragile, unreliable, and more than a decade out of date. Thus, we also built a second pipeline using Python scripts in Blender to load in the remeshed models and render the contours using Blender's version of Freestyle. Both pipelines had pros and cons, with the original library Freestyle having the ability to be integrated directly into our C++ code, but with the Blender pipeline proving significantly more robust on unusual meshes.

Finally, we rendered the model a second time using the watercolor rendering technique described in the second paper. Special care was taken to avoid the "shower door" effect so the results would appear convincing even as part of an animation. The technique was ultimately implemented using GLSL shaders and then composited with the contour renders.

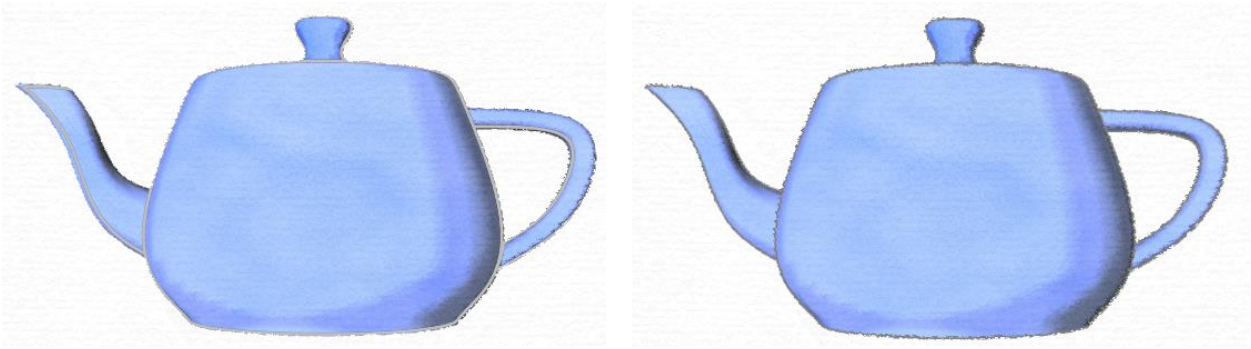
This pipeline – remeshing, rendering contours, rendering watercolors, compositing, and then saving the result – was repeated many times to produce a series of frames that we could then play as an animation.

For examples of our project's output, see the videos in this playlist:
https://www.youtube.com/playlist?list=PLlsx-oXgq2liqhqn7sEyV2Os6qH3bt-hP

Example Pictures



The teapot after remeshing. In this picture, the remeshing was done as if facing the teapot from its handle. The view was then rotated to show the line of changed triangles added at the division between front and back facing triangles.



Left: Final composite render of a teapot using Blender contours.
Right: Final composite render of a teapot using raw Freestyle contours.

Contours on the remeshed Stanford Bunny rendered using Blender.