Real-time Cel Shading: Final Report

Bidipta Sarkar
bidiptas@stanford.edu

Summary:

I created a Cel-Shading look in OpenGL based off of the shader code in assignment 3. I also implemented shadow maps for directional light (sunlight). Finally, I downloaded an obj from Sketchfab to show how it looks on a real character.

Cel-Shading:

Cel shading is a common stylized rendering technique that mimics the style of 2D animation. I wanted my version of cel shading to match my personal style of 2D drawing, which includes having (1) distinct palettes for areas in shadow and light, (2) sharp specular highlights, and (3) outlines of varying thickness.

Diffuse Palettes:
In standard Phong shading, the intensity of the diffuse color is proportional to the (clamped) dot product of the surface normal and the light direction. To have separate colors for light and dark, I just set a threshold for an acceptable dot product. Specifically, if the value is below 0.25, the diffuse intensity is a quarter of the "light" intensity.

Specular Highlights:
Like the diffuse case, I just set a threshold for the specular component. I also had to ensure that the specular highlight does not overlap with the shadow area. I noticed that when normal mapping is enabled, we get an interesting cross-hatching look, which I thought made the render feel more like a drawing.

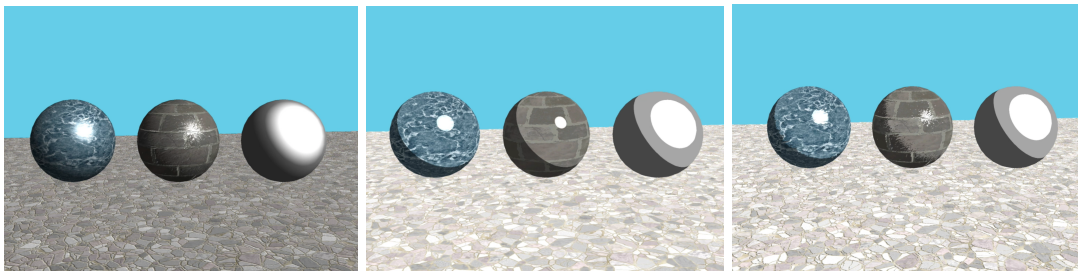

Figure 1: (Left) Phong; (Center) Shading without normals; (Right) Shading with normals

Outlines:

There are many techniques for outlines in Cel-shading. I started by trying the typical wireframe method, which involves translating and rendering the back-faces of objects multiple times before doing the final true rendering. However, this technique resulted in uniform, thin outlines, which didn't feel right.

Instead, I took inspiration from the Fresnel equations, which describe how dielectrics become more reflective as viewing angles become nearly tangent to a surface. Whenever I do 2D drawings, I realized that I would ink in strokes wherever the viewer's gaze would be grazing off of the object, which includes contours within the figure. After implementing the Shlick approximation of the Fresnel equations, I found that I could continue simplifying until I reached a very simple threshold equation based on the dot product of the direction to the camera and the normal.

Unfortunately, this technique has some limitations that I had to work around. We assume a continuously changing surface for these "nearly tangent" expressions, but this would result in artifacts on planar surfaces that are already nearly tangent to the camera. Thankfully, this was not very visible for high-poly or generally smooth meshes, so I just excluded the floor from the border-creation algorithm.

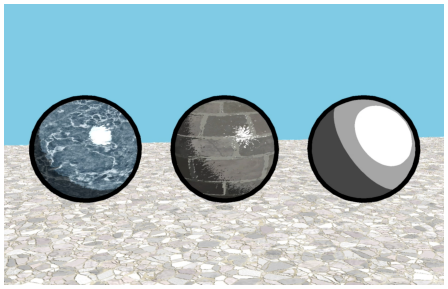

Figure 2: Full Cel-Shading on Spheres

Sunlight Shadows:

The second major component of this project was generating shadows for directional light. Unlike spotlights, directional light does not have a fixed field of view or a fixed location. If we directly represent the directional light as a very distant and powerful spotlight, the shadow map would have a very low resolution. However, we know that an infinitely far projection can be modeled as an orthographic projection, so we can just replace the spotlight's perspective NDC matrix with an orthographic one. After ensuring that the entire scene's bounding box is within the normalized frame, we can proceed with the standard shadow mapping algorithm.

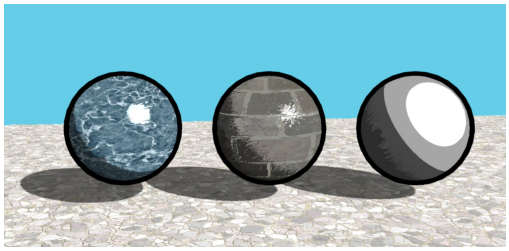

Figure 3: Full Cel-Shading on Spheres with Shadows

Final Scene:

I downloaded a free model of Link, the protagonist of *The Legend of Zelda: Ocarina of Time*, from Sketchfab. I also used Blender to re-export the model into individual pieces so the textures would be preserved in the render. I also used a free dirt texture from Poliigon.

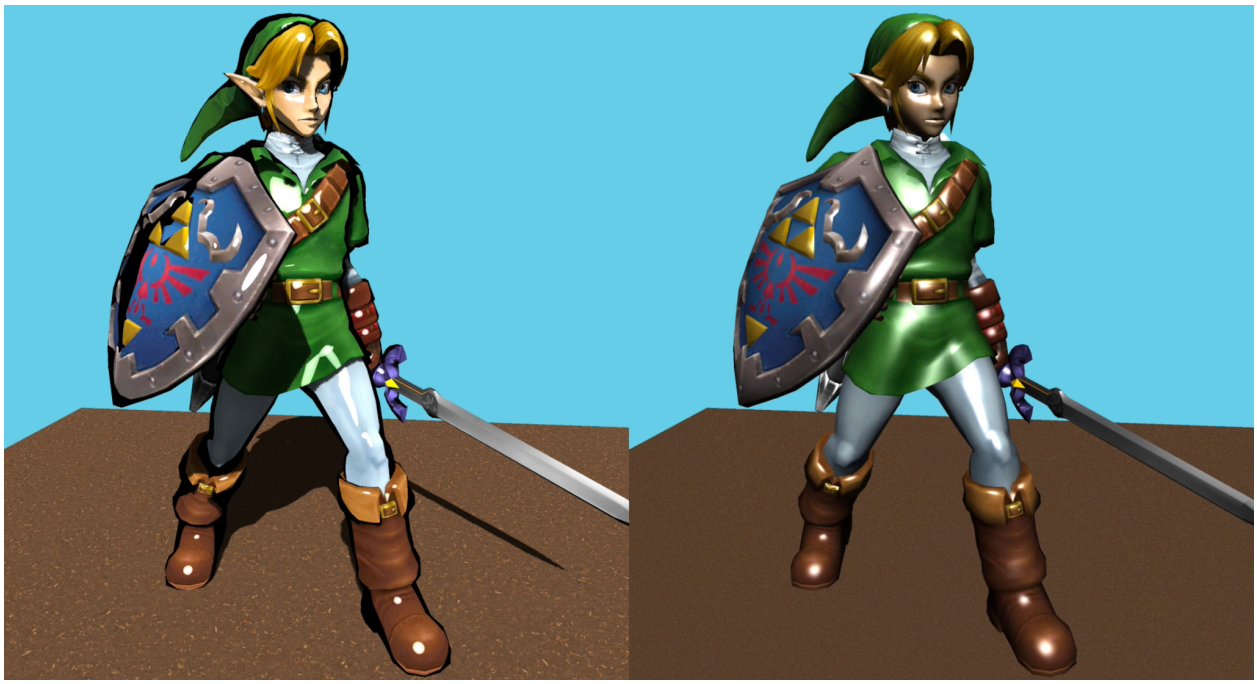

Figure 4: (Left) Cel-Shaded Link + Shadows; (Right) Phong-Shaded Link

I think the final Cel-shaded look is very visually appealing. We can see how the boots and pants have a thick border, but different areas have a thinner border. The shadow mapping worked well overall, since different parts of Link cast shadows on other parts, like the hair on the face. Compared to the simple phong-shading, the cel-shaded look really captures the feeling of a 2D drawing.

All the files for this render are inside the folder celshading/media/zelda. This folder also contains link_cel.json and link_phong.json which can be used to recreate this scene.

References:

[1] "Ocarina of Time Link" (https://skfb.ly/DNps) by projectmgame is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

- Link model for the final render

[2] Poliigon Ground Dirt Forest 014

- Dirt texture and normal for final render

[3]https://raulreyesfinalproject.files.wordpress.com/2012/12/dissertation_cell-shading-raul_reyes_luque.pdf

- Helpful resource for understanding Cel-shading techniques