

Gooch/Toon Shading with Silhouettes

James Cabot

6.4400: Introduction to Computer Graphics, MIT

December 2025

1 Motivation

The motivation for the project is clear: **technical illustrations need clear shape depiction more than photorealism**. Clear shape depiction will add detail to the model's edges and shading such that all of the relevant information from the object can be seen. Photorealism, the depiction of objects as they would appear in real life, is not as important to the rendering of technical illustrations because the details of the image are the value of the rendering, not the representation of the light and colors of the real world.

Phong shading is a shading option nearer the photorealistic side of this spectrum. It has dark shadows and bright highlights, like a real-world image, which can obscure the edges of rendered objects. For example, see **Figure 1**, a teapot rendered with Phong shading. The shadows obscure some of the detail of the spout, and the highlights obscure some of the detail of the handle. This is the problem that the project is addressing.

The solution is Gooch shading. This option introduces gradient tones to keep the edges and interiors of the rendered objects legible. It attempts to keep more of the detail of the technical rendering, details that Phong shading obscures. Therefore, the project goal is to create an interactive non-photorealistic renderer (NPR) for explanatory 3D visualization.

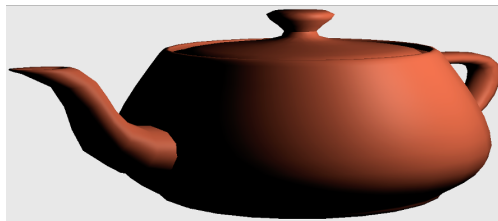


Figure 1: Teapot rendered with Phong shading. Shadows obscure spout detail; highlights wash out handle.

2 Background

The origin of Gooch shading is from SIGGRAPH in 1998, where Gooch et al. presented their initial publication of gradient shading for technical illustrations [1]. This is now called Gooch shading. The paper replaced luminance with color tones, it added mid-tone interiors, and it added black silhouettes to preserve shape. Compared with Phong shading, the shadows and highlights are displayed better.

Technical foundation: Gooch et al. identified four key principles of technical illustration:

- **Black edge lines** (silhouettes, boundaries, discontinuities) drawn first
- **Mid-tone interiors** using cool→warm hue shifts to indicate surface orientation
- **White specular highlights** from single light source
- **No cast shadows** (form > scene lighting)

Gooch’s core equation replaces Phong’s luminance term:

$$\text{Phong: } I = k_d \cdot \max(0, \hat{n} \cdot \hat{l}) \quad (\text{full dynamic range}) \quad (1)$$

$$\begin{aligned} \text{Gooch: } t &= \left(\frac{\hat{n} \cdot \hat{l} + 1}{2} \right)^2 \\ k_{\text{cool}} &= k_{\text{blue}} + \alpha \cdot k_d, \quad k_{\text{warm}} = k_{\text{yellow}} + \beta \cdot k_d \\ I &= (1 - t)k_{\text{cool}} + tk_{\text{warm}} \quad (\text{mid-tones only}) \end{aligned} \quad (2)$$

where $k_{\text{blue}} = (0, 0, b)$, $k_{\text{yellow}} = (y, y, 0)$, $\alpha, \beta \in [0, 1]$.

This is important in technical rendering because the value of the rendering is in the virtual image and the information it portrays. Gooch et al.’s mission to improve the abilities for technical renderings and this information availability generated this Gooch shading, gradient-based visualization.

3 Approach

The first step was to create a working code model with Phong shading. To actually visualize this, the Stanford Teapot and Stanford Bunny models were created. They are simple, well-known models for rendering and visualization, and they focus the project on the shading and rendering implementation, not the models themselves. Phong shading works by creating rendering, shading, and material components and using a frag model to specify the light values at each triangle. Phong shading uses diffuse and specular shading to calculate the color of each triangle in the rendering. Once this initial workflow was proofed, the ability to render an object was solidified. Then, Gooch shading can be added.

This has a very similar overall structure. Gooch shading starts with the same rendering, shading, and material components, and it generates its triangle colors from a frag model as well. However, this frag model is slightly different. Gooch shading mixes the warm and cool colors that are set at either end of the gradient and calculates their impact on each triangle using the normal and light vectors of the triangle. The result is the Gooch shading color gradient seen in **Figure 2**. The triangles closest to the light are orange, and the triangles obscured from the light are blue.

Based on this idea, there are two extensions for the rendering. Toon shading has the same ideas as Gooch shading, but it quantizes the tone parameter and renders a number of zones across the model, each of one uniform color. This rendering is used to exaggerate the forms of objects, which makes it great for games or explanatory diagrams. It is likely recognizable in **Figure 3**. The functional implementation is almost identical to Gooch. The difference is the calculation of the triangle color in the frag model. The model makes use of the floor function to group triangles together and give them all the same color. The result is a color gradient in the same direction, but one that is very segmented instead of smooth like Gooch shading.

The final extension is silhouette rendering. This is a separate geometry pass from the shader rendering. It uses backface culling and extrusion in another frag model to show some shadows on the rendered object. It is an addition to the previously established framework for rendering each of the three shaders. The main use is to enhance the 3-D capabilities of the object by showing its edges. While Gooch is an improvement over Phong shading, some of the colors of the gradient blend together, so full detail can still be lacking. The silhouette extension adds some more details to the object that further differentiate the part of the object that might have the same color and thus blend together.

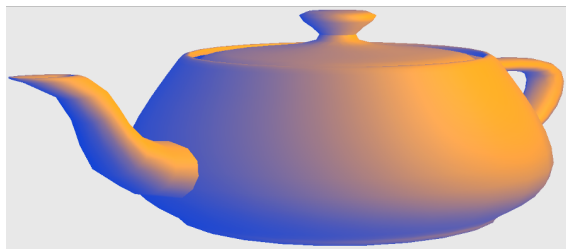


Figure 2: Teapot rendered with Gooch shading, cool-to-warm gradient.

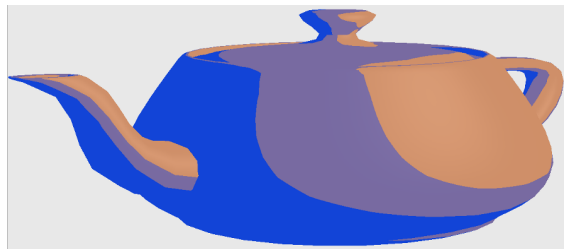


Figure 3: Teapot rendered with Toon shading (3 quantized bands).

4 Results

The results are six options of object rendering. The interactive program has Phong, Gooch, and Toon shading, and a toggle for the silhouettes. The silhouettes do not improve the Phong shader at all, but they significantly help the detail of the Gooch and Toon shaders. The program also includes a toggle between the models, switching between teapot and bunny. Below are the twelve total outputs of the program (Figures 4–15).

Technical observations confirming the analysis:

- **Phong:** Shadows obscure teapot spout details (Fig. 4) and bunny ear geometry (Fig. 10). Silhouettes blend into dark regions.
- **Gooch:** Cool→warm gradient ($k_{\text{cool}} = (0, 0.2, 0.8)$, $k_{\text{warm}} = (1, 0.6, 0.1)$, biased $t = ((N \cdot L + 1)/2)^2$) reveals all surface details.
- **Toon:** 3-band quantization ($t_q = \lfloor t \cdot 3 \rfloor / 3$) exaggerates form for diagrams.
- **Silhouettes:** Hull geometry (backface culling + 0.02-unit extrusion) dramatically enhances 3D perception.

5 Conclusion

This project successfully implements Gooch shading for technical illustration, directly matching the 1998 SIGGRAPH paper’s core requirements: mid-tone cool→warm interiors that preserve shape information and explicit black silhouettes for edge definition. The baseline Phong shader confirms the problem (crushed shadows and highlights obscure details), while Gooch and its extensions demonstrate the solution.

Key achievements:

- Interactive renderer with keyboard controls (1/2/3 shading modes, O toggle outlines, M model switch)
- Toon quantization extension creates discrete 3-band shading for stylized diagrams
- Hull-based silhouettes via backface culling + extrusion work reliably across complex topology
- Real-time performance (60+ FPS) suitable for interactive technical visualization

The results validate Gooch et al.’s insight: when silhouettes are explicit, low dynamic-range mid-tone shading suffices for clear form communication. Silhouettes dramatically enhance Gooch/Toon (spout/ear details pop), but add little value to Phong (already crushed blacks).

Future extensions include runtime Toon band adjustment (+/- keys), crease-aware outlines (sharp mesh edges), rim lighting for edge glow, and multi-light support. The full source code for this project is available at: <https://github.com/Jimmyc776/gooch-toon-shading-silhouette>.

References

- [1] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of SIGGRAPH*, pages 447–452, 1998. Available at <https://users.cs.northwestern.edu/~ago820/SIG98/gooch98.pdf>.

6 Appendix

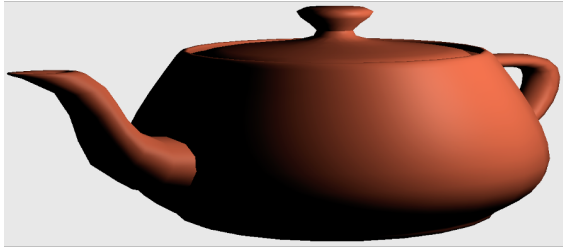


Figure 4: Teapot: Phong shading, no silhouettes.

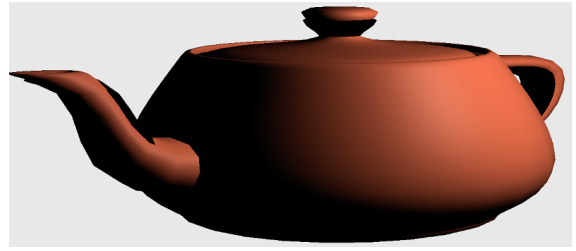


Figure 5: Teapot: Phong shading, yes silhouettes.

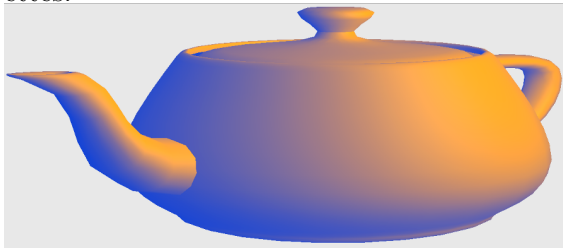


Figure 6: Teapot: Gooch shading, no silhouettes.

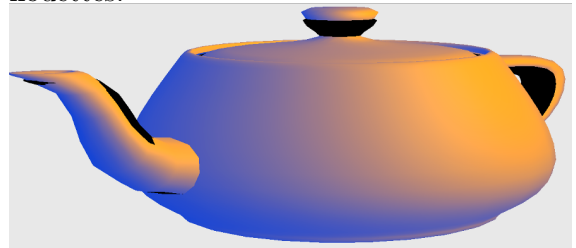


Figure 7: Teapot: Gooch shading, yes silhouettes.

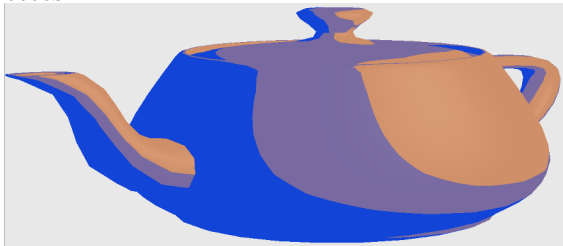


Figure 8: Teapot: Toon shading, no silhouettes.

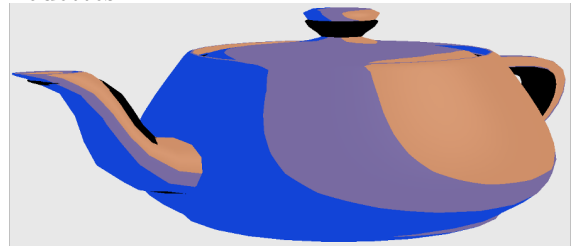


Figure 9: Teapot: Toon shading, yes silhouettes.

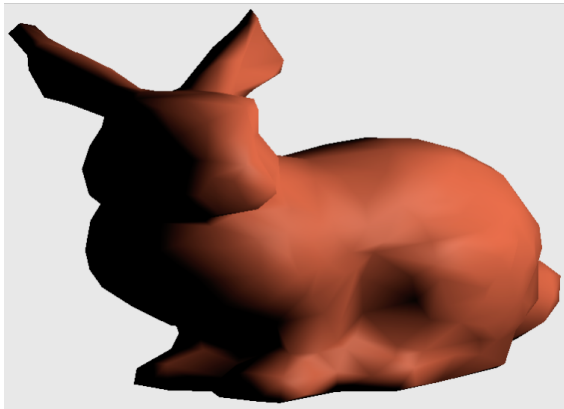


Figure 10: Bunny: Phong shading, no silhouettes.

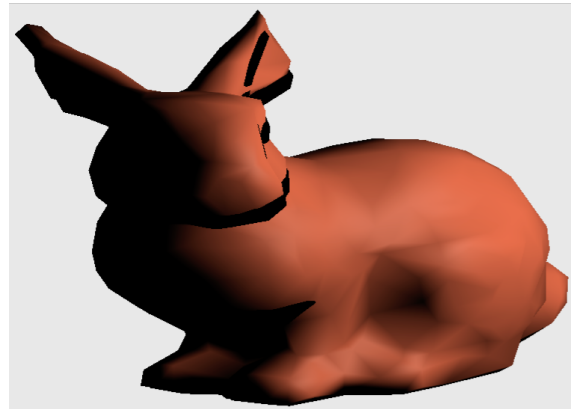


Figure 11: Bunny: Phong shading, yes silhouettes.

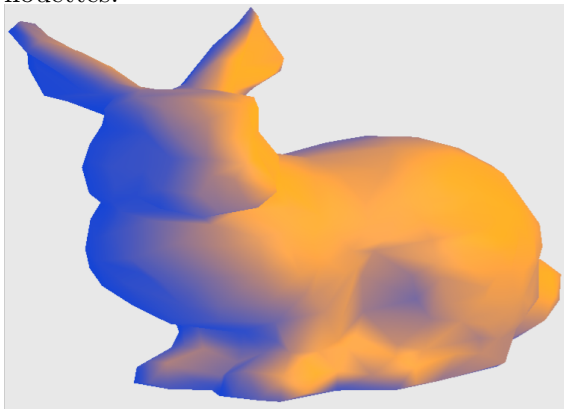


Figure 12: Bunny: Gooch shading, no silhouettes.

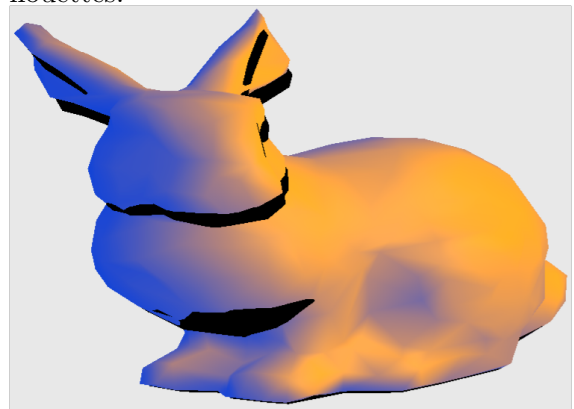


Figure 13: Bunny: Gooch shading, yes silhouettes.

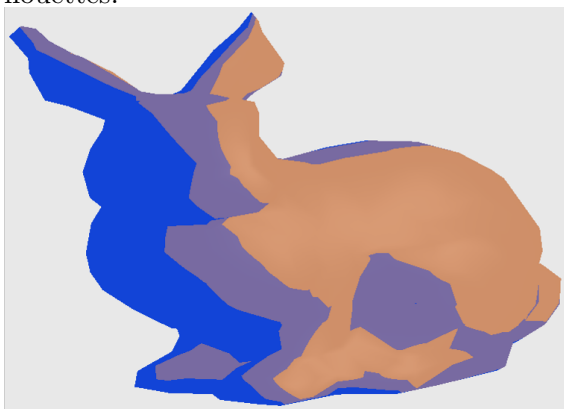


Figure 14: Bunny: Toon shading, no silhouettes.

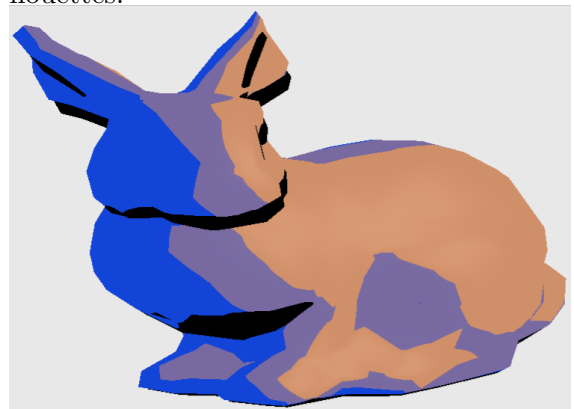


Figure 15: Bunny: Toon shading, yes silhouettes.