A Non-Photorealistic Lighting Model For Automatic Technical Illustration

Amy Gooch **Bruce Gooch** **Peter Shirley** **Elaine Cohen**

Department of Computer Science
University of Utah
http://www.cs.utah.edu/

Abstract

Phong-shaded 3D imagery does not provide geometric information of the same richness as human-drawn technical illustrations. A non-photorealistic lighting model is presented that attempts to narrow this gap. The model is based on practice in traditional technical illustration, where the lighting model uses both luminance and changes in hue to indicate surface orientation, reserving extreme lights and darks for edge lines and highlights. The lighting model allows shading to occur only in mid-tones so that edge lines and highlights remain visually prominent. In addition, we show how this lighting model is modified when portraying models of metal objects. These illustration methods give a clearer picture of shape, structure, and material composition than traditional computer graphics methods.

CR Categories: I.3.0 [Computer Graphics]: General; I.3.6 [Computer Graphics]: Methodology and Techniques.

Keywords: illustration, non-photorealistic rendering, silhouettes, lighting models, tone, color, shading

1 Introduction

The advent of photography and computers has not replaced artists, illustrators, or draftsmen, despite rising salaries and the decreasing cost of photographic and computer rendering technology. Almost all manuals that involve 3D objects, e.g., a car owner's manual, have illustrations rather than photographs. This lack of photography is present even in applications where aesthetics are a side-issue, and communication of geometry is the key. Examining technical manuals, illustrated textbooks, and encyclopedias reveals illustration conventions that are quite different from current computer graphics methods. These conventions fall under the umbrella term *technical illustrations*. In this paper we attempt to automate some of these conventions. In particular, we adopt a shading algorithm based on cool-to-warm *tones* such as shown in the non-technical image in Figure 1. We adopt this style of shading to ensure that black silhouettes and edge lines are clearly visible which is often not the case when they are drawn in conjunction with traditional computer graphics shading. The fundamental idea in this paper is that when silhouettes and other edge lines are explicitly drawn, then very low dynamic range shading is needed for the interior. As artists have discovered, adding a somewhat artificial hue shift to shading helps imply shape without requiring a large dynamic range. This hue shift can interfere with precise albedo perception, but this is not a major concern in technical illustration where the communication of shape and form are valued above realism. In Section 2 we review previous computer graphics work, and conclude that little has been done to produce shaded technical drawings. In Section 3 we review the common technical illustration practices. In Section 4 we describe how we have automated some of these practices. We discuss future work and summarize in Section 5.

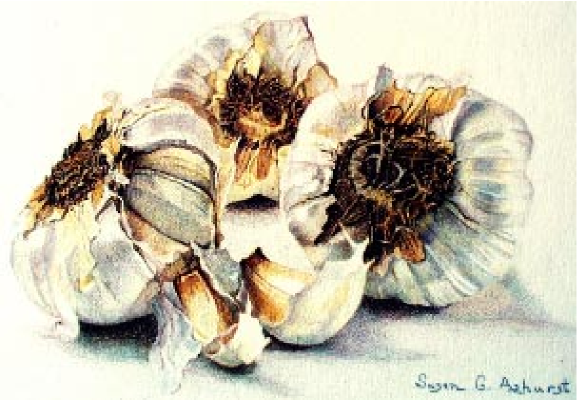

Figure 1: *The non-photorealistic cool (blue) to warm (tan) transition on the skin of the garlic in this non-technical setting is an example of the technique automated in this paper for technical illustrations. Colored pencil drawing by Susan Ashurst.*

2 Related Work

Computer graphics algorithms that imitate non-photographic techniques such as painting or pen-and-ink are referred to as *non-photorealistic rendering* (NPR). The various NPR methods differ greatly in style and visual appearance, but are all closely related to conventional artistic techniques (e.g., [6, 8, 10, 13, 14, 16, 20, 26]). An underlying assumption in NPR is that artistic techniques developed by human artists have intrinsic merit based on the evolutionary nature of art. We follow this assumption in the case of technical illustration.

NPR techniques used in computer graphics vary greatly in their level of abstraction. Those that produce a loss of detail, such as semi-randomized watercolor or pen-and-ink, produce a very high level of abstraction, which would be inappropriate for most technical illustrations. Photorealistic rendering techniques provide little abstraction, so photorealistic images tend to be more confusing than less detailed human-drawn technical illustrations. Technical illustrations occupy the middle ground of abstraction, where the important three-dimensional properties of objects are accented while extraneous detail is diminished or eliminated. Images at any level of abstraction can be aesthetically pleasing, but this is a side-effect rather than a primary goal for technical illustration. A rationale for using abstraction to eliminate detail from an image is that, unlike the case of 3D scene perception, the image viewer is not able to use motion, accommodation, or parallax cues to help deal with visual complexity. Using abstraction to simplify images helps the user overcome the loss of these spatial cues in a 2D image.

In computer graphics, there has been little work related to technical illustration. Saito and Takahashi [19] use a variety of techniques to show geometric properties of objects, but their images do not follow many of the technical illustration conventions. Seligmann and Feiner present a system that automatically generates explanation-based drawings [21]. Their system focuses primarily on what to draw, with secondary attention to visual style. Our work deals primarily with visual style rather than layout issues, and thus there is little overlap with Seligmann and Feiner's system, although the two methods would combine naturally. The work closest to our own was presented by Dooley and Cohen [7] who employ a user-defined hierarchy of components, such as line width, transparency, and line end/boundary conditions to generate an image. Our goal is a simpler and more automatic system, that imitates methods for line and color use found in technical illustrations. Williams also developed similar techniques to those described here for non-technical applications, including some warm-to-cool tones to approximate global illumination, and drawing conventions for specular objects [25].

3 Illustration Techniques

Based on the illustrations in several books, e.g. [15, 18], we conclude that illustrators use fairly algorithmic principles. Although there are a wide variety of styles and techniques found in technical illustration, there are some common themes. This is particularly true when we examine color illustrations done with air-brush and pen. We have observed the following characteristics in many illustrations:

- edge lines, the set containing surface boundaries, silhouettes, and discontinuities, are drawn with black curves.
- matte objects are shaded with intensities far from black or white with warmth or coolness of color indicative of surface normal; a single light source provides white highlights.
- shadowing is not shown.
- metal objects are shaded as if very anisotropic.

We view these characteristics as resulting from a hierarchy of priorities. The edge lines and highlights are black and white, and provide a great deal of shape information themselves. Several studies in the field of perception have concluded that subjects can recognize 3D objects at least as well, if not better, when the edge lines (contours) are drawn versus shaded or textured images [1, 3, 5, 22]. However, when shading is added in addition to edge lines, more information is provided only if the shading uses colors that are visually distinct from both black and white. This means the dynamic range available for shading is extremely limited. In most technical illustrations, shape information is valued above precise reflectance information, so hue changes are used to indicate surface orientation rather than reflectance. This theme will be investigated in detail in the next section.

A simple low dynamic-range shading model is consistent with several of the principles from Tufte's recent book [23]. He has a case-study of improving a computer graphics animation by lowering the contrast of the shading and adding black lines to indicate direction. He states that this is an example of the strategy of *the smallest effective difference*:

> *Make all visual distinctions as subtle as possible, but still clear and effective.*

Tufte feels that this principle is so important that he devotes an entire chapter to it. The principle provides a possible explanation of why cross-hatching is common in black and white drawings and rare in colored drawings: colored shading provides a more subtle, but adequately effective, difference to communicate surface orientation.

4 Automatic Lighting Model

All of the characteristics from Section 3 can be automated in a straightforward manner. Edge lines are drawn in black, and highlights are drawn using the traditional exponential term from the Phong lighting model [17]. In Section 4.1, we consider matte objects and present reasons why traditional shading techniques are insufficient for technical illustration. We then describe a low dynamic range artistic tone algorithm in Section 4.2. Next we provide an alogrithm to approximate the anisotropic appearance of metal objects, described in Section 4.3. We provide approximations to these algorithms using traditional Phong shading in Section 4.4.

4.1 Traditional Shading of Matte Objects

In addition to drawing edge lines and highlights, we need to shade the surfaces of objects. Traditional diffuse shading sets luminance proportional to the cosine of the angle between light direction and surface normal:

$$I = k_d k_a + k_d \ \max\left(0, \hat{\mathbf{l}} \cdot \hat{\mathbf{n}}\right) \tag{1}$$

where I is the RGB color to be displayed for a given point on the surface, k_d is the RGB diffuse reflectance at the point, k_a is the RGB ambient illumination, $\hat{\mathbf{l}}$ is the unit vector in the direction of the light source, and $\hat{\mathbf{n}}$ is the unit surface normal vector at the point. This model is shown for $k_d = 1$ and $k_a = 0$ in Figure 3. This unsatisfactory image hides shape and material information in the dark regions. Additional information about the object can be provided by both highlights and edge lines. These are shown alone in Figure 4 with no shading. We cannot effectively add edge lines and highlights to Figure 3 because the highlights would be lost in the light regions and the edge lines would be lost in the dark regions.

To add edge lines to the shading in Equation 1, we can use either of two standard heuristics. First we could raise k_a until it is large enough that the dim shading is visually distinct from the black edge lines, but this would result in loss of fine details. Alternatively, we could add a second light source, which would add conflicting highlights and shading. To make the highlights visible on top of the shading, we can lower k_d until it is visually distinct from white. An image with hand-tuned k_a and k_d is shown in Figure 5. This is the best achromatic image using one light source and traditional shading. This image is poor at communicating shape information, such as details in the claw nearest the bottom of the image. This part of the image is colored the constant shade $k_d k_a$ regardless of surface orientation.

4.2 Tone-based Shading of Matte Objects

In a colored medium such as air-brush and pen, artists often use both hue and luminance (greyscale intensity) shifts. Adding blacks and whites to a given color results in what artists call *shades* in the case of black, and *tints* in the case of white. When color scales are

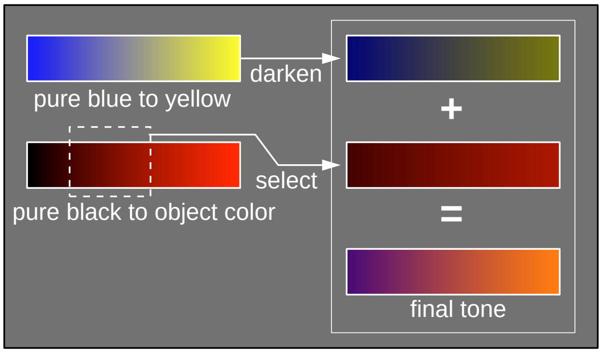

Figure 2: *How the tone is created for a pure red object by summing a blue-to-yellow and a dark-red-to-red tone.*

created by adding grey to a certain color they are called *tones* [2]. Such tones vary in hue but do not typically vary much in luminance. When the complement of a color is used to create a color scale, they are also called tones. Tones are considered a crucial concept to illustrators, and are especially useful when the illustrator is restricted to a small luminance range [12]. Another quality of color used by artists is the *temperature* of the color. The temperature of a color is defined as being warm (red, orange, and yellow), cool (blue, violet, and green), or temperate (red-violets and yellow-greens). The depth cue comes from the perception that cool colors recede while warm colors advance. In addition, object colors change temperature in sunlit scenes because cool skylight and warm sunlight vary in relative contribution across the surface, so there may be ecological reasons to expect humans to be sensitive to color temperature variation. Not only is the temperature of a hue dependent upon the hue itself, but this advancing and receding relationship is effected by proximity [4]. We will use these techniques and their psychophysical relationship as the basis for our model.

We can generalize the classic computer graphics shading model to experiment with tones by using the cosine term ($\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}$) of Equation 1 to blend between two RGB colors, k_{cool} and k_{warm}:

$$I = \left(\frac{1+\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}}{2}\right) k_{cool} + \left(1 - \frac{1+\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}}{2}\right) k_{warm} \qquad (2)$$

Note that the quantity $\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}$ varies over the interval $[-1, 1]$. To ensure the image shows this full variation, the light vector $\hat{\mathbf{l}}$ should be perpendicular to the gaze direction. Because the human vision system assumes illumination comes from above [9], we chose to position the light up and to the right and to keep this position constant.

An image that uses a color scale with little luminance variation is shown in Figure 6. This image shows that a sense of depth can be communicated at least partially by a hue shift. However, the lack of a strong cool to warm hue shift and the lack of a luminance shift makes the shape information subtle. We speculate that the unnatural colors are also problematic.

In order to automate this hue shift technique and to add some luminance variation to our use of tones, we can examine two extreme possibilities for color scale generation: blue to yellow tones and scaled object-color shades. Our final model is a linear combination of these techniques. Blue and yellow tones are chosen to insure a cool to warm color transition regardless of the diffuse color of the object.

The blue-to-yellow tones range from a fully saturated blue: $k_{blue} = (0, 0, b), b \in [0, 1]$ in RGB space to a fully saturated yellow: $k_{yellow} = (y, y, 0), y \in [0, 1]$. This produces a very sculpted but unnatural image, and is independent of the object's diffuse reflectance k_d. The extreme tone related to k_d is a variation of diffuse shading where k_{cool} is pure black and $k_{warm} = k_d$. This would look much like traditional diffuse shading, but the entire object would vary in luminance, including where $\hat{\mathbf{l}} \cdot \hat{\mathbf{n}} < 0$. What we would really like is a compromise between these strategies. These transitions will result in a combination of tone scaled object-color and a cool-to-warm undertone, an effect which artists achieve by combining pigments. We can simulate undertones by a linear blend between the blue/yellow and black/object-color tones:

$$\begin{aligned} k_{cool} &= k_{blue} + \alpha k_d \\ k_{warm} &= k_{yellow} + \beta k_d \end{aligned} \qquad (3)$$

Plugging these values into Equation 2 leaves us with four free parameters: b, y, α, and β. The values for b and y will determine the strength of the overall temperature shift, and the values of α and β will determine the prominence of the object color and the strength of the luminance shift. Because we want to stay away from shading which will visually interfere with black and white, we should supply intermediate values for these constants. An example of a resulting tone for a pure red object is shown in Figure 2.

Substituting the values for k_{cool} and k_{warm} from Equation 3 into the tone Equation 2 results in shading with values within the middle luminance range as desired. Figure 7 is shown with $b = 0.4$, $y = 0.4$, $\alpha = 0.2$, and $\beta = 0.6$. To show that the exact values are not crucial to appropriate appearance, the same model is shown in Figure 8 with $b = 0.55$, $y = 0.3$, $\alpha = 0.25$, and $\beta = 0.5$. Unlike Figure 5, subtleties of shape in the claws are visible in Figures 7 and 8.

The model is appropriate for a range of object colors. Both traditional shading and the new tone-based shading are applied to a set of spheres in Figure 9. Note that with the new shading method objects retain their "color name" so colors can still be used to differentiate objects like countries on a political map, but the intensities used do not interfere with the clear perception of black edge lines and white highlights.

4.3 Shading of Metal Objects

Illustrators use a different technique to communicate whether or not an object is made of metal. In practice illustrators represent a metallic surface by alternating dark and light bands. This technique is the artistic representation of real effects that can be seen on milled metal parts, such as those found on cars or appliances. Milling creates what is known as "anisotropic reflection." Lines are streaked in the direction of the axis of minimum curvature, parallel to the milling axis. Interestingly, this visual convention is used even for smooth metal objects [15, 18]. This convention emphasizes that realism is not the primary goal of technical illustration.

To simulate a milled object, we map a set of twenty stripes of varying intensity along the parametric axis of maximum curvature. The stripes are random intensities between 0.0 and 0.5 with the stripe closest to the light source direction overwritten with white. Between the stripe centers the colors are linearly interpolated. An object is shown Phong-shaded, metal-shaded (with and without edge lines), and metal-shaded with a cool-warm hue shift in Figure 10. The metal-shaded object is more obviously metal than the Phong-shaded image. The cool-warm hue metal-shaded object is not quite as convincing as the achromatic image, but it is more visually consistent with the cool-warm matte shaded model of Section 4.2, so it is useful when both metal and matte objects are shown together. We note that our banding algorithm is very similar to the technique Williams applied to a clear drinking glass using image processing [25].

4.4 Approximation to new model

Our model cannot be implemented directly in high-level graphics packages that use Phong shading. However, we can use the Phong lighting model as a basis for approximating our model. This is in the spirit of the non-linear approximation to global illumination used by Walter et al. [24]. In most graphics systems (e.g. OpenGL) we can use negative colors for the lights. We can approximate Equation 2 by two lights in directions $\hat{\mathbf{l}}$ and $-\hat{\mathbf{l}}$ with intensities $(k_{warm} - k_{cool})/2$ and $(k_{cool} - k_{warm})/2$ respectively, and an ambient term of $(k_{cool} + k_{warm})/2$. This assumes the object color is set to white. We turn off the Phong highlight because the negative blue light causes jarring artifacts. Highlights could be added on systems with accumulation buffers [11].

This approximation is shown compared to traditional Phong shading and the exact model in Figure 11. Like Walter et al., we need different light colors for each object. We could avoid these artifacts by using accumulation techniques which are available in many graphics libraries.

Edge lines for highly complex objects can be generated interactively using Markosian et al.'s technique [14]. This only works for polygonal objects, so higher-order geometric models must be tessellated to apply that technique. On high-end systems, image-processing techniques [19] could be made interactive. For metals on a conventional API, we cannot just use a light source. However, either environment maps or texture maps can be used to produce alternating light and dark stripes.

5 Future Work and Conclusion

The shading algorithm presented here is exploratory, and we expect many improvements are possible. The most interesting open ended question in automatic technical illustrations is how illustration rules may change or evolve when illustrating a scene instead of single objects, as well as the practical issues involved in viewing and interacting with 3D technical illustrations. It may also be possible to automate other application-specific illustration forms, such as medical illustration.

The model we have presented is tailored to imitate colored technical drawings. Once the global parameters of the model are set, the technique is automatic and can be used in place of traditional illumination models. The model can be approximated by interactive graphics techniques, and should be useful in any application where communicating shape and function is paramount.

Acknowledgments

Thanks to Bill Martin, David Johnson, Brian Smits and the members of the University of Utah Computer Graphics groups for help in the initial stages of the paper, to Richard Coffey for helping to get the paper in its final form, to Bill Thompson for getting us to reexamine our interpretation of why some illustration rules might apply, to Susan Ashurst for the sharing her wealth of artistic knowledge, to Dan Kersten for valuable pointers into the perception literature, and to Jason Herschaft for the dinosaur claw model. This work was originally inspired by the talk by Jane Hurd in the SIGGRAPH 97 panel on medical visualization. This work was supported in part by DARPA (F33615-96-C-5621) and the NSF Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219). All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

References

[1] Irving Biederman and Ginny Ju. Surface versus Edge-Based Determinants of Visual Recognition. *Cognitive Psychology*, 20:38–64, 1988.

[2] Faber Birren. *Color Perception in Art*. Van Nostrand Reinhold Company, 1976.

[3] Wendy L. Braje, Bosco S. Tjan, and Gordon E. Legge. Human Efficiency for Recognizing and Detecting Low-pass Filtered Objects. *Vision Research*, 35(21):2955–2966, 1995.

[4] Tom Browning. *Timeless Techniques for Better Oil Paintings*. North Light Books, 1994.

[5] Chris Christou, Jan J. Koenderink, and Andrea J. van Doorn. Surface Gradients, Contours and the Perception of Surface Attitude in Images of Complex Scenes. *Perception*, 25:701–713, 1996.

[6] Cassidy J. Curtis, Sean E. Anderson, Kurt W. Fleischer, and David H. Salesin. Computer-Generated Watercolor. In *SIGGRAPH 97 Conference Proceedings*, August 1997.

[7] Debra Dooley and Michael F. Cohen. Automatic Illustration of 3D Geometric Models: Surfaces. *IEEE Computer Graphics and Applications*, 13(2):307–314, 1990.

[8] Gershon Elber and Elaine Cohen. Hidden Curve Removal for Free-Form Surfaces. In *SIGGRAPH 90 Conference Proceedings*, August 1990.

[9] E. Bruce Goldstein. *Sensation and Perception*. Wadsworth Publishing Co., Belmont, California, 1980.

[10] Paul Haeberli. Paint By Numbers: Abstract Image Representation. In *SIGGRAPH 90 Conference Proceedings*, August 1990.

[11] Paul Haeberli. The Accumulation Buffer: Hardware Support for High-Quality Rendering. *SIGGRAPH 90 Conference Proceedings*, 24(3), August 1990.

[12] Patricia Lambert. *Controlling Color: A Practical Introduction for Designers and Artists*, volume 1. Everbest Printing Company Ltd., 1991.

[13] Peter Litwinowicz. Processing Images and Video for an Impressionistic Effect. In *SIGGRAPH 97 Conference Proceedings*, August 1997.

[14] L. Markosian, M. Kowalski, S. Trychin, and J. Hughes. Real-Time Non-Photorealistic Rendering. In *SIGGRAPH 97 Conference Proceedings*, August 1997.

[15] Judy Martin. *Technical Illustration: Materials, Methods, and Techniques*, volume 1. Macdonald and Co Publishers, 1989.

[16] Barbara J. Meier. Painterly Rendering for Animation. In *SIGGRAPH 96 Conference Proceedings*, August 1996.

[17] Bui-Tuong Phong. Illumination for Computer Generated Images. *Communications of the ACM*, 18(6):311–317, June 1975.

[18] Tom Ruppel, editor. *The Way Science Works*, volume 1. MacMillan, 1995.

[19] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3D Shapes. In *SIGGRAPH 90 Conference Proceedings*, August 1990.

[20] Mike Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable Textures for Image-Based Pen-and-Ink Illustration. In *SIGGRAPH 97 Conference Proceedings*, August 1997.

[21] Doree Duncan Seligmann and Steven Feiner. Automated Generation of Intent-Based 3D Illustrations. In *SIGGRAPH 91 Conference Proceedings*, July 1991.

[22] Bosco S. Tjan, Wendy L. Braje, Gordon E. Legge, and Daniel Kersten. Human Efficiency for Recognizing 3-D Objects in Luminance Noise. *Vision Research*, 35(21):3053–3069, 1995.

[23] Edward Tufte. *Visual Explanations*. Graphics Press, 1997.

[24] Bruce Walter, Gun Alppay, Eric P. F. Lafortune, Sebastian Fernandez, and Donald P. Greenberg. Fitting Virtual Lights for Non-Diffuse Walkthroughs. In *SIGGRAPH 97 Conference Proceedings*, pages 45–48, August 1997.

[25] Lance Williams. Shading in Two Dimensions. *Graphics Interface '91*, pages 143–151, 1991.

[26] Georges Winkenbach and David H. Salesin. Computer Generated Pen-and-Ink Illustration. In *SIGGRAPH 94 Conference Proceedings*, August 1994.

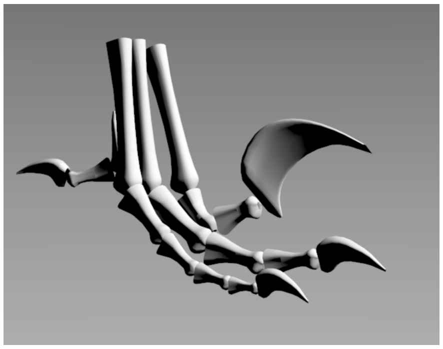

Figure 3: Diffuse shaded image using Equation 1 with $k_d = 1$ and $k_a = 0$. Black shaded regions hide details, especially in the small claws; edge lines could not be seen if added. Highlights and fine details are lost in the white shaded regions.

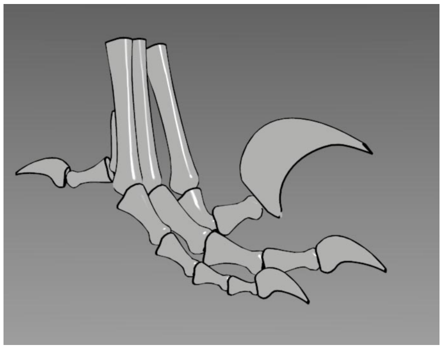

Figure 4: Image with only highlights and edges. The edge lines provide divisions between object pieces and the highlights convey the direction of the light. Some shape information is lost, especially in the regions of high curvature of the object pieces. However, these highlights and edges could not be added to Figure 3 because the highlights would be invisible in the light regions and the silhouettes would be invisible in the dark regions.

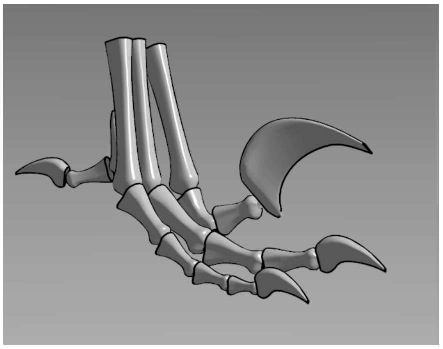

Figure 5: Phong shaded image with edge lines and $k_d = 0.5$ and $k_a = 0.1$. Like Figure 3, details are lost in the dark grey regions, especially in the small claws, where they are colored the constant shade of $k_d k_a$ regardless of surface orientation. However, edge lines and highlights provide shape information that was gained in Figure 4, but couldn't be added to Figure 3.

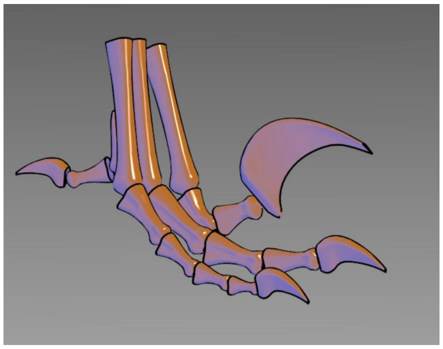

Figure 6: Approximately constant luminance tone rendering. Edge lines and highlights are clearly noticeable. Unlike Figures 3 and 5 some details in shaded regions, like the small claws, are visible. The lack of luminance shift makes these changes subtle.

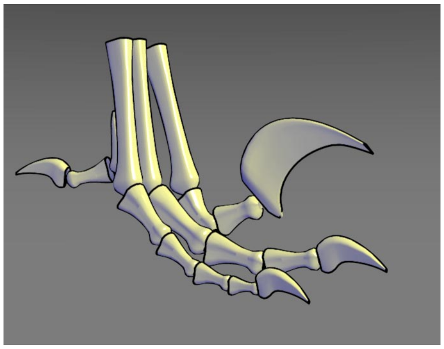

Figure 7: Luminance/hue tone rendering. This image combines the luminance shift of Figure 3 and the hue shift of Figure 6. Edge lines, highlights, fine details in the dark shaded regions such as the small claws, as well as details in the high luminance regions are all visible. In addition, shape details are apparent unlike Figure 4 where the object appears flat. In this figure, the variables of Equation 2 and Equation 3 are: $b = 0.4$, $y = 0.4$, $\alpha = 0.2$, $\beta = 0.6$.

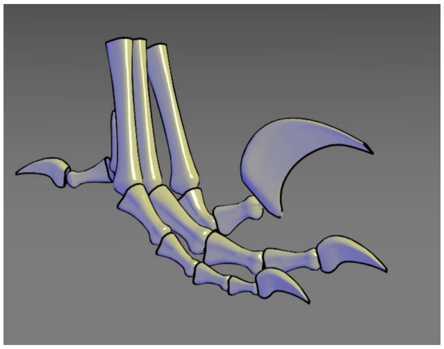

Figure 8: Luminance/hue tone rendering, similar to Figure 7 except $b = 0.55$, $y = 0.3$, $\alpha = 0.25$, $\beta = 0.5$. The different values of b and y determine the strength of the overall temperature shift, where as α and β determine the prominence of the object color, and the strength of the luminance shift.

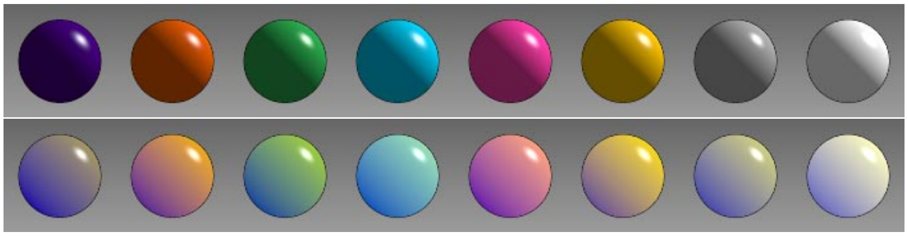

Figure 9: Top: Colored Phong-shaded spheres with edge lines and highlights. Bottom: Colored spheres shaded with hue and luminance shift, including edge lines and highlights. Note: In the first Phong shaded sphere (violet), the edge lines disappear, but are visible in the corresponding hue and luminance shaded violet sphere. In the last Phong shaded sphere (white), the highlight vanishes, but is noticed in the corresponding hue and luminance shaded white sphere below it. The spheres in the second row also retain their “color name”.

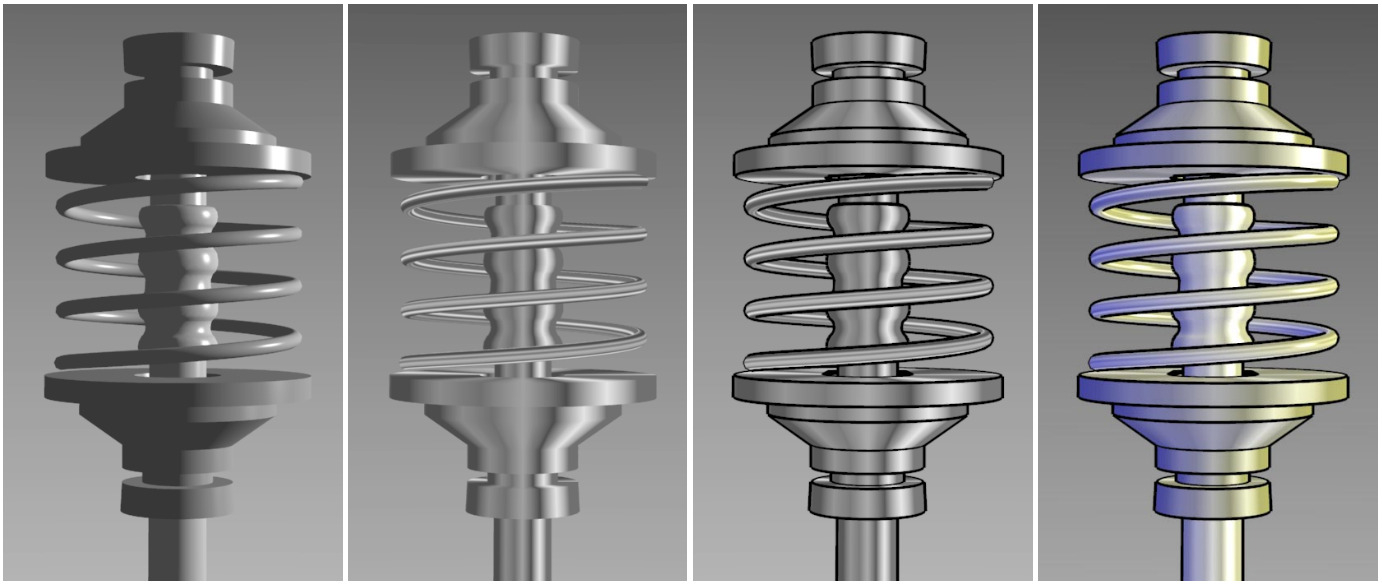

Figure 10: Left to Right: a) Phong shaded object. b) New metal-shaded object without edge lines. c) New metal-shaded object with edge lines. d) New metal-shaded object with a cool-to-warm shift.

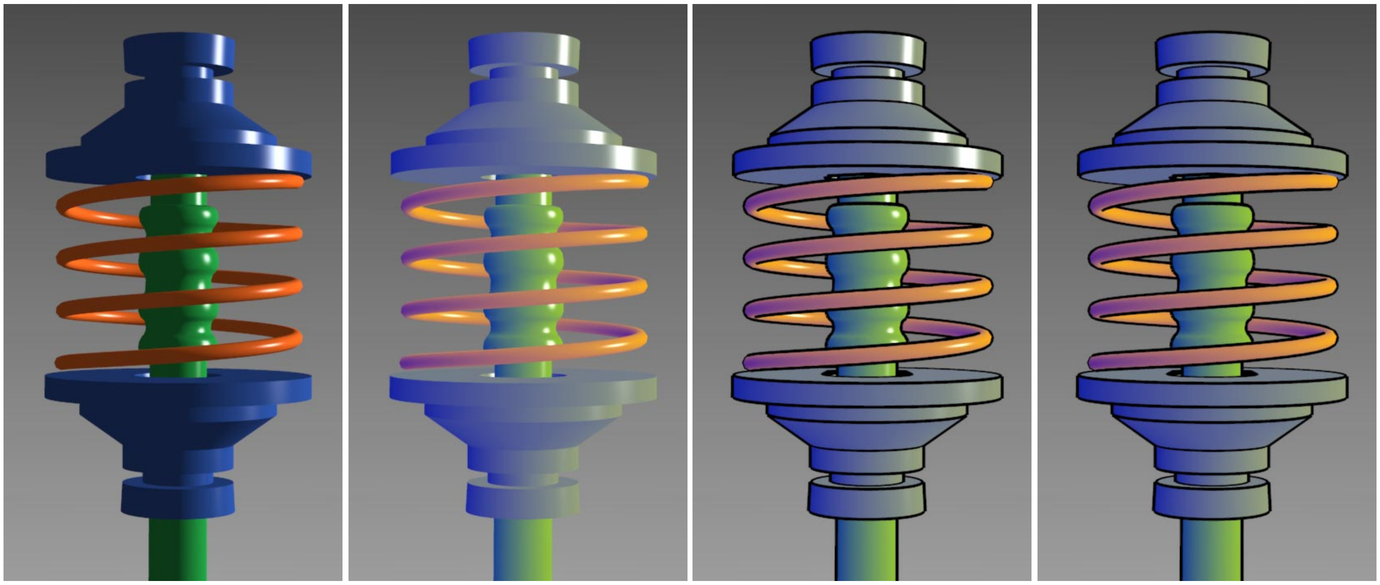

Figure 11: Left to Right: a) Phong model for colored object. b) New shading model with highlights, cool-to-warm hue shift, and without edge lines. c) New model using edge lines, highlights, and cool-to-warm hue shift. d) Approximation using conventional Phong shading, two colored lights, and edge lines.