

# Color Styling Tools

Clement Shimizu & Gary W. Meyer; Department of Computer Science and Engineering  
Digital Technology Center, University of Minnesota; Minneapolis, MN

## Abstract

Creative tools are proposed that allow color stylists to take advantage of their training in the art and design fields. A simple reflection model is employed that has the minimum number of free parameters required to design solid and metallic color finishes from conceptualization to fabrication. The parameters correspond to color specification terms familiar to designers such as face color, flop color, travel, and gloss. We demonstrate how the reflection model can also be used to develop effective interfaces for color stylists. We create a virtual mood board that allows direct selection of the reflection model parameters from pictures. We also develop an image based BRDF tweaker for adjusting color appearance directly on a 3D object.

## Introduction

Industrial designers, interior designers, and architects all make extensive use of exemplars when they choose colors for a new product. They collect material samples from suppliers, they select illustrations from fashion magazines, and they take pictures of objects with interesting surface finishes. These exemplars are organized and displayed on “mood boards” (see Figure 1) that allow the designer to compare the colors and make final selections. The designer uses the mood board as a reference point when communicating their intentions to others and applying the design. Traditional hue, saturation, and brightness (HSB) color organization systems are used, after the fact, to provide a name and a specification for the final color.

The existing computer graphic tools for color simulation and selection do not readily facilitate the color design process used in these traditional design fields. Computer graphic reflection models have been developed with the goal of accurately portraying subtle color appearance effects. These models have grown increasingly complex, and they include physical parameters for which aesthetic designers have no intuition. Numerous computer graphic HSB color systems have been proposed, but they are not able by themselves to convey the spatial aspects that separate one color appearance from another. In addition, in the traditional design process, exemplars and sketches are used more often than HSB color systems to ideate color.

In this paper we propose a set of computer graphic tools to facilitate color design in the traditional aesthetic design fields. We employ a reflection model that covers the widest range of color appearances encountered by designers including solid colors, metallic colors, and the glossiness of these colors. The reflection model is constructed to have the minimum number of free parameters and to select these parameters so that they correspond to color specification terms familiar to designers such as face color, flop color, travel, and gloss. The model is defined in a way that allows it to be used as a manufacturing specification for the final color.



**Figure 1.** Designers often use mood boards to propose the selection of colors and materials. They might start out collecting images from fashion magazines or other visually appealing artifacts like dried leaves or butterfly wings that can be pinned onto a mood board. As the designer's vision develops, they rearrange and refine the images on the board. Finally, the designer uses the mood board when discussing their concepts with others and as a reference when implementing the design. The art director of a style and fashion magazine was invited to participate in the user study. She proposed “Orange Ball of Paris” that appeals to a young and fashionable audience.

We also demonstrate how the reflection model can be used to develop effective interfaces for color designers. We create a virtual mood board that allows direct selection of the reflection model parameters from pictures. We also describe a novel color design interface for tweaking color and appearance directly on a 3D surface.

## Relevant work

There is a variety of background work in providing artistic and design oriented controls over computer graphic reflection models. Design galleries presents renderings with variations of computer graphic parameters as a interface for tweaking scene parameters in [11]. Ngan [12] created image driven navigation systems for specifying the parameters to various BRDF models. Pelacini created a psychophysically-based BRDF model that aided the specification of gloss parameters in [13]. Kautz [8] allows an artist to create a bitmap texture that represents the shape and color of a reflectance distribution. Khan introduced a system for doing image space material editing in [9]. This paper focused on the rendering system that could apply new material designs to existing images.

An intuitive and artistic interface for painting BRDFs was presented in [3]. The interface provides the ability to create a BRDF by positioning and manipulating highlights on a spherical canvas. A mapping between painted highlights and specular lobes was created for an extended Ward model. This software is constrained to a point light source and a spherical canvas.

Sloan [19] developed a novel technique for capturing 3D non-photorealistic shading models from 2D artwork. The interface has a tool for selecting a region of a 2D source image to approximate an illuminated sphere that is later used to render 3D models. This allows for unconstrained lighting and shape, but does not pull pure BRDF information. The information collected convolves lighting with the BRDF, similar to a prefiltered environment map.

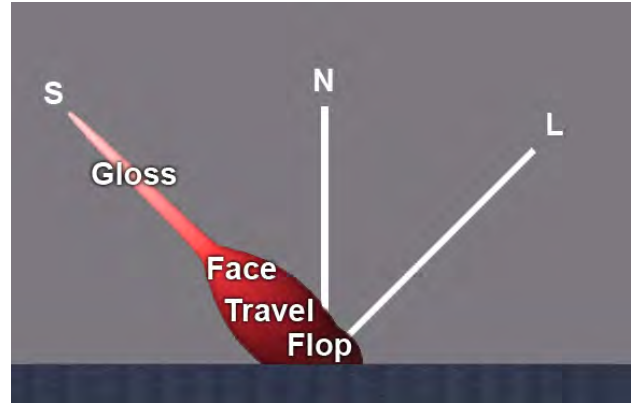
Outside of our own work, there has been a variety of research on realistic rendering of optical phenomena relevant to the design of automotive paints. Rendering of interference and wave based optical phenomena was covered in [4, 7, 20]. Image based measurement techniques was applied to automotive paints in [6, 16].

## Face/flop/travel/gloss reflection model

This section describes the reflection model. First, the variables of the reflection model are defined, along with how they control the physical appearance of the resulting material. Since these variables cannot be directly related to graphics rendering, they are linked to a parametric representation that has been previously tied to a realtime rendering engine [17]. This reflection model is connected to the physical world through measurement tools and rapid color prototyping systems.

### Face/flop/travel/gloss form

We describe a reflection model that has the following parameters: face, flop, travel, and gloss. Figure 2 gives a visualization of how the parameters of this reflection model correlate to the shape of the BRDF. “Face” is the color at  $15^\circ$  off specular angle and is represented as a Lab tristimulus value. This angle establishes the directionally diffuse color while avoiding the specular highlight resulting from the first surface coatings. “Flop” is the color at far from specular and is represented as a Lab tristimulus value. The particular angle that flop is associated with is the variable travel. Travel represents how color changes with viewing direction. This is intended to be the color of the diffuse portion of the material at as close to a specular angle as possible. If the face and flop colors are identical, the color is a solid color and therefore travel does not alter the appearance of the material.



**Figure 2.** A diagram illustrating how the four components of the metallic reflection model relate to the shape of the BRDF lobe. The incoming light  $L$  is at  $45^\circ$  from the surface normal  $N$ . The specular reflection direction is labeled  $S$ .

In many computer graphic reflection models, gloss is represented by two parameters, one representing the color and magnitude of the specular lobe and one representing the width of the specular lobe. In the Phong model this would be a color  $K_s$  and a scalar specular exponent. In our reflection model gloss is a single scalar number represented by ASTM standard gloss. This one number controls both the size and scale of the first surface reflection’s specular lobe. [22] showed a mapping between ASTM standard gloss and parameters to Phong and Ward specular lobes. This single value sacrifices some degrees of freedom for ease of use in representing first surface reflection.

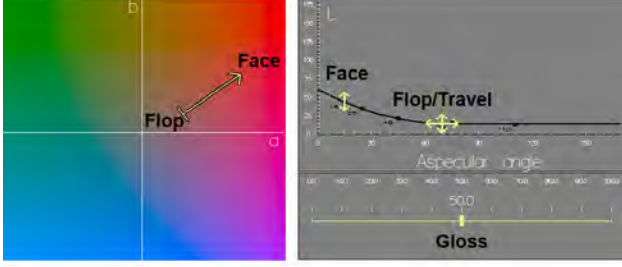
Although our model has connections to and from the physical world, because it is an appearance model, it stands in contrast to physically based reflection models. A layered shader model based on the microscopic structure and optical properties of the constituents pigments, metallic flakes, binders, and substrates was proposed in [5]. This bottom up approach does not facilitate color styling because of its highly technical nature.

### Parametric form

The face/flop/travel/gloss reflection model has limited use outside of design work, and is unsuitable for direct rendering. However, it is easily converted to a parametric form for rendering purposes. Figure 3 shows how the face/flop/travel/gloss reflection model fits on top of the parametric form of the reflection model. This parametric form has appeared in a slightly different form in [18].

For metallic paints, the variation in appearance with reflectance angle, or travel, occurs at a very slow rate. In a study of metallic automobile paints, Alman showed that only three tristimulus measurements are necessary to describe travel [1]. Moreover, a second degree polynomial is all that is needed to interpolate the measurements in CIE Lab color space. We therefore convert the face/flop/travel/gloss model parameters into a second degree polynomial for  $L$ ,  $a$ , and  $b$ .

In this section we work through the mathematical details only for the  $L$  polynomial curve. To compute the complete BRDF, the same equations are performed on the  $a$  and  $b$  curves as well. In



**Figure 3.** An interface for directly manipulating face, flop, travel, and gloss. The  $L$  vs. aspecular angle plot illustrates how the parametric form interpolates face at  $15^\circ$  and flop at  $\theta_{clamp}$ .

the equations below,  $L_a$ ,  $L_b$ , and  $L_c$  refer to the three components of the quadratic polynomial representing  $L$ .

$$L(\theta) = L_a \theta^2 + L_b \theta + L_c \quad (1)$$

$$L_{clamped}(\theta) = L(\min(\theta, \theta_{clamp})) \quad (2)$$

Setting the derivative of the quadratic polynomial to be zero and solving for  $\theta_{clamp}$ , we find the clamp point needed to evaluate the clamped quadratic polynomials.

$$\frac{dL(\theta_{clamp})}{d\theta} = 0 = 2L_a \theta_{clamp} + L_b \quad (3)$$

$$\theta_{clamp} = \frac{-L_b}{2L_a} \quad (4)$$

To fit the curve, Alman [1] suggested that three tristimulus measurements were necessary: one near specular, one near grazing, and one angle in between. Others confirmed his results in [15,21].

To solve for the three unknowns of the quadric polynomial, three equations are needed. Although normally the color at three angles is needed to define the three equations, we can use a representation that has only two angles: face and flop. Since flop represents the quadric curve's apex and the derivative at this point must be zero, we now have three equations to find the three unknowns  $L_a$ ,  $L_b$ , and  $L_c$ .

$$L(15^\circ) = L_a * (15^\circ)^2 + L_b 15^\circ + L_c \quad (5)$$

$$L(\theta_{clamp}) = L_a \theta_{clamp}^2 + L_b \theta_{clamp} + L_c \quad (6)$$

$$\frac{dL(\theta_{clamp})}{d\theta} = 0 = 2L_a \theta_{clamp} + L_b \quad (7)$$

Solving for  $L_a$ ,  $L_b$ , and  $L_c$  therefore depends on knowing the face color ( $L(15^\circ)$ ), the flop color ( $L(\theta_{clamp})$ ), and the travel (determined by  $\theta_{clamp}$ ). To convert back is simple. First evaluate Equation 4 to find travel as  $\theta_{clamp}$  in degrees. Then evaluate Equation 2 at  $15^\circ$  degrees and  $\theta_{clamp}$  to find face and flop color as  $Lab(15^\circ)$  and  $Lab(\theta_{clamp})$ . The final solutions to the equations are shown below:

$$L_a = \frac{L(15^\circ) - L(\theta_{clamp})}{(15^\circ - \theta_{clamp})^2} \quad (8)$$

$$L_b = -2L_a \theta_{clamp} \quad (9)$$

$$L_c = L(\theta_{clamp}) + L_a \theta_{clamp}^2 \quad (10)$$

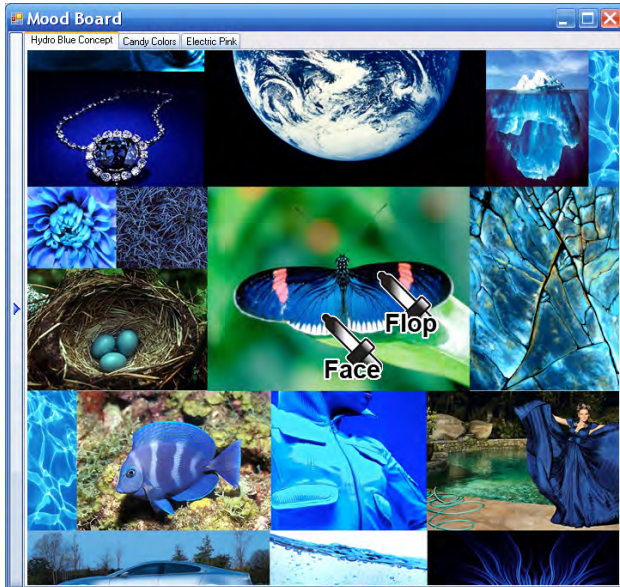


**Figure 4.** Inspired by the sporty fashion trends of California's Orange County, a mood board proposes the color "Hydrogen Blue in the OC" for a hydrogen powered concept car. It reflects the creator's acute insight into style, aesthetics, and cultural cues that are relevant to the design concept.

## Virtual mood board

Designers frequently seek inspiration from images and other visual artifacts. Their creative work flow often starts with something called a mood board [2]. Mood boards can also be called collages, story boards or inspiration boards. A mood board typically consists of a large piece of foam core or cardboard. A designer uses thumbtacks to arrange images cut from magazines, concept art, and other visual artifacts on the mood board. In the aesthetic design fields, mood boards are used extensively for material selection. The process might start out with a stylist collecting images from a stock art web site or from a fashion magazine. Materials such as a piece of leather from a glove, a belt buckle, or a seashell might also be placed on the mood board. In Figure 4 a real mood board was created to select color appearances for a hydrogen concept vehicle.

Similar to the process of using a real mood board, the digital work flow starts with the stylist collecting and arranging various digital images on the virtual mood board as shown on the left of Figure 5. Next, they select the face, flop, and gloss of the reflection model from the mood board using eyedroppers and visualize the results on an object as shown on the right. Once satisfied with the design, they export the color as standard aspecular measurements to be rapidly prototyped using a paint formulation system.



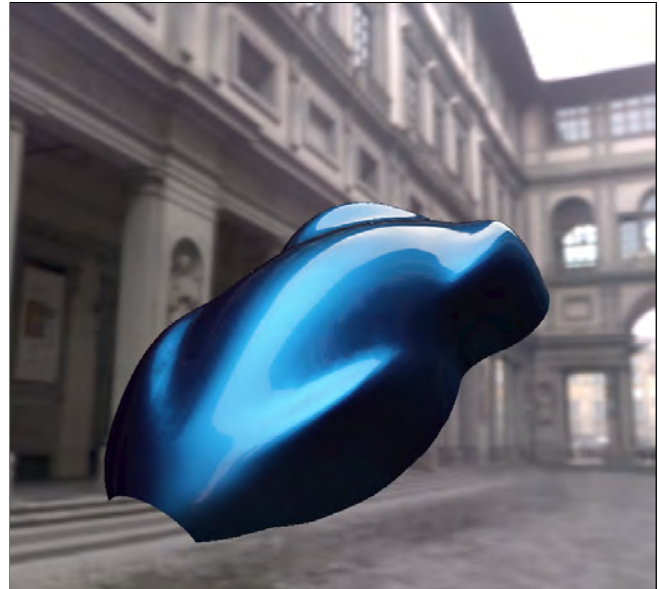
**Figure 5.** A designer was instructed to use the virtual mood board software to create a color relevant to a hydrogen concept vehicle. The designer arranges inspirational digital images for material concepts on the mood board (left). They select the  $15^\circ$  face color and  $\theta_{clamp}$  flop color using eyedroppers. The rendering on the right is of a material seeded by the colors in the butterfly's wings.

## Interface

The virtual mood board's user interface is designed to mimic the work flow of a real mood board. The user creates separate boards for different projects using the virtual mood board's tabbed interface. In the same way a stylist arranges images cut from magazines on a real mood board using thumb tacks, the stylist arranges digital images on the virtual mood board using the mouse to drag them around. The scroll function of the mouse is mapped to zooming in and out of the image under the cursor. Once the stylist has some images arranged on the mood board, they can now use the system to select and visualize materials.

Although it would be ideal to be able to sample full BRDF measurements from the images found on the mood board, this is not yet possible. Obtaining BRDFs from materials in photographs is an inverse rendering problem. A reflected light field is a convolution of the incident lighting and the BRDF [14]. Therefore, to reconstruct a BRDF from a photograph, the incident lighting and scene geometry are needed. There are a variety of techniques for photographic acquisition of BRDFs. These techniques need scene geometry, controlled or known lighting, and/or multiple photographs [10, 23, 24]. Because the images found on a mood board do not have information about scene geometry and incident lighting, inverse rendering techniques cannot be used for obtaining BRDF information.

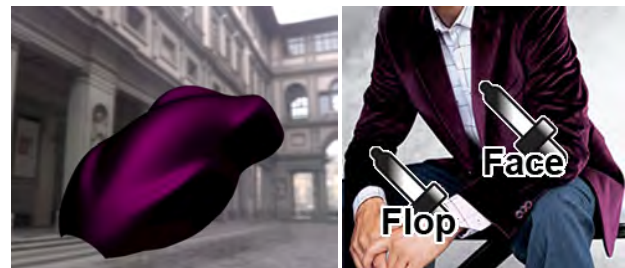
The reflection model presented in the previous section allows the virtual mood board to use a simple but effective interface for selecting material properties from images. Remember, the parameters of the reflection model that need to be set are face, flop, travel, and gloss. The parameters are set using eyedroppers to directly sample color from the mood board. One eyedropper is labeled face and is used for picking  $Lab(15^\circ)$ . One eyedropper is labeled flop and is used for choosing  $Lab(\theta_{clamp})$ . A special gloss eyedropper selects gloss. A slider provides control of travel ( $\theta_{clamp}$ ).



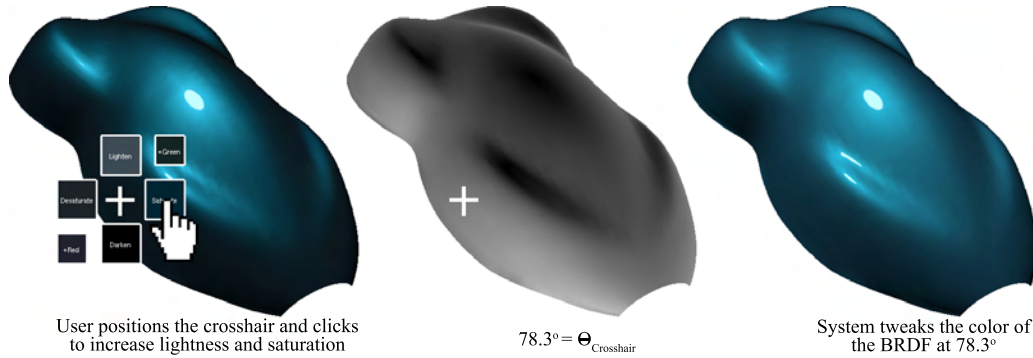
## Face/flop eyedropper

The metaphor of using an eyedropper to select a color from an image is best known from Adobe Photoshop, although other commercial products use the same eyedropper paradigm for color selection. Although the use of eyedroppers is not new, a dramatic improvement in usability is achieved because the eyedroppers are paired with the virtual mood board and the face/flop reflection model.

The user drags each eyedropper over images in the mood board using the mouse cursor. As the user drags the eyedropper over the images the program samples over a small range of pixels under the mouse cursor to provide a steady color selection. When the user releases the mouse cursor the eyedropper rests over the drop point. The mood board software converts the face and flop color sampled from the eyedroppers (as well as the travel from the sliders) to the parametric form of the reflection model. The system uses the rendering engine described in [17] to display a preview of the color as the user drags the eyedropper around the mood board. The parametric form of the reflection model is used (along with pre-captured environment maps to this rendering engine) to display the new colors as they are designed in real time.



**Figure 6.** Although the spatial distribution of velvet is not captured from this image, the system still duplicates the color appearance.



**Figure 7.** The original color looks a little too dull on the bottom left edge of the object. The stylist positions the crosshair over this area and increases saturation and brightness (left). A shader renders an specular angle to an off screen buffer to assist mapping adjustments from image-space to BRDF space (center). By indexing into the buffer with the crosshair coordinates, the system calculates that the color should be adjusted at  $78.3^\circ$  (right).

One interesting use of eyedroppers is to pull color inspiration from one type of material to seed another type of material. These material types can be widely differing. For instance, Figure 6 shows a designed metallic automotive paint that is the color of a velvet sport coat. While the metallic reflection model cannot represent the spatial reflectance of velvet, the eyedroppers can still capture the particular shades of that color.

There are some improvements to the system that make it more useable. Since retroreflective colors are uncommon and not well represented by the face/flop/travel reflection model our software assumes that the face component of the reflection model is never darker than the flop component. The system defaults to automatically swap the eyedroppers if the user is selecting a face color that is darker than the flop color or vice versa. This option can be disabled to style retroreflective colors. Another interface improvement is to allow the eyedropper to sample parts of the screen outside the mood board. This allows the designer to sample from not only image inside the mood board, but also directly from websites, videos, 2D and 3D color collections, Quicktime VRs, and any interactive 3D application.

### Image based BRDF tweaker

Color stylists often want to make minor modifications to automotive color designs. They may realize that the color needs tweaking only after seeing the color on a three dimensional surface like a car mockup. They may want the color on the hood to be a little brighter or the color on the doors to be a little less saturated. Since the car is painted a uniform color, their request must be interpreted as modifications to the face and/or flop of a color that will affect the entire surface. A stylist with a deeper understanding of the behavior of automotive finishes would be able to make adjustments to the face, flop, and travel to indirectly get the desired appearance of the automobile. A tool that allows stylists to specify color adjustments directly at particular positions on the body of a car would provide more precise control over the genesis of the color's appearance.

The image based BRDF tweaker is a color design interface for tweaking the color and appearance of the reflection model directly on a 3D object. With the BRDF tweaker, we are extending the ideas originally proposed in [3] for use on any shape to provide rich, complete control over the parameters of the face/flop/travel/gloss reflection model.

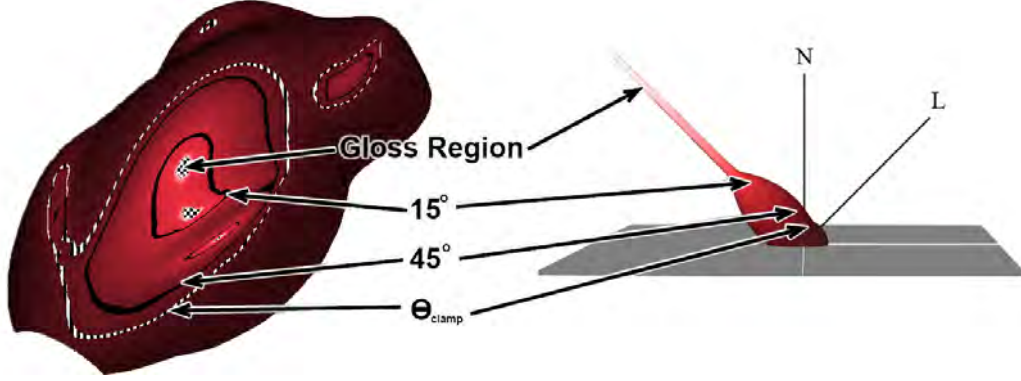
We first describe the mode of operation and then describe the how the software achieves it using an off-screen buffer to map image space screen coordinates to the parameters of the BRDF.

### Tools and mode of operation

The BRDF tweaker is integrated into a simple point light rendering engine. The BRDF tweaker is context sensitive and changes the display and mode of operation depending on a number of factors. The tool consists of a draggable *crosshair* widget for tweaking color at specific areas on the object and a heads up display(HUD) for visualizing and adjusting spatial aspects of the BRDF.

When the draggable crosshair widget is positioned on top of the 3D object, the crosshair widget expands to show color manipulation tools and the HUD is disabled. The color manipulation tools let the user drag a crosshair to positions on the object and directly adjust the lightness, saturation, and hue at that particular area of the object. As the crosshair is moved across the surface of the object, the widget samples the aspecular angle under the crosshair. The color of the BRDF at the particular aspecular angle is computed. Surrounding the crosshair are buttons that show variations of the color of the BRDF under the crosshair. Above and below the crosshair are buttons that show lighter and darker versions of the color. To the left and right are increased and decreased saturation (also called chroma), and diagonally are hue rotations. When a button is pressed, the color is adjusted at that position, it updates the reflection model using the technique described in the section titled "adjust color", and the change is reflected in the entire image. For example, if the crosshair is over the *face* region (as shown on the bottom left of Figure 7) of the 3D object, then adjustments only change the *face* of the BRDF while leaving the *flop* color untouched. If the crosshair is in between the face and the flop and the saturate button is clicked, it will adjust the saturation color at both face and flop to varying degrees.

When the draggable crosshair widget is dragged off of the object, the crosshair widget contracts and hides the color manipulation tools and the HUD is enabled. The HUD shows a visualization of the aspecular lighting angles as a topology map. This is very useful for understanding exactly how the color plays or travels across the object. The heads up display is driven by a hardware shader that computes the angle between the reflected light and view vector (the aspecular angle). The left side of Figure 8 shows



**Figure 8.** The left shows the heads up display over an auto speed shape. Specific aspecular angles in image space are correlated to positions on a graph of the BRDF for this illustration (right).

the heads up display over a three dimensional shape designed by the automotive industry to help evaluate color appearance. Pixels being rendered by the shader are highlighted if their aspecular angle is within a few degrees of  $15^\circ$ ,  $45^\circ$ , or  $\theta_{clamp}$ . The topology of the ring representing  $\theta_{clamp}$  changes depending on the rate of travel of the BRDF. Additionally, the gloss region is displayed with a checkerboard. In this mode, the stylist can drag the ring representing  $\theta_{clamp}$  to end at a particular position on the 3D object. Also, if the user clicks on the edge of the specular highlight they can drag it to adjust the width of the specular highlight to affect the gloss. Finally, if they click in the center of the specular highlight then they can drag the specular reflection to a new point on the surface of the object. This is an easy goal driven way of adjusting the light position.

### Making adjustments

The BRDF tweaker is powered by a series of shaders that render extra information about the scene's reflectance in image space into off screen buffers. Point lighting allows us to compute an exact mapping between pixels on the screen and lighting angles or viewing angles of the BRDF. The principle parameter of the reflection model is aspecular angle. A shader computes this and stores it in an off screen buffer. This needs to be computed when the scene, view, or lighting changes. It does not need to be recomputed if the only modification is to the BRDF.

### Adjust color

If the crosshair is on top of the object on any spot other than a specular highlight, adjustments are made to specific angles of the metallic portion of the reflection model depending on exactly where the crosshair is positioned. If the user has clicked a color adjustment button, the face and flop of the reflection model is adjusted by a delta in lightness ( $\Delta L$ ), saturation/chroma ( $\Delta C$ ), or hue ( $\Delta H$ ) as shown in Equations 11, 12, and 13 respectively depending on which button they picked. These equations are applied to *both* face and flop to create temporary colors, marked by a single prime symbol,  $Lab'(15^\circ)$  and  $Lab'(\theta_{clamped})$ .

$$Lab' = (L + \Delta L, a, b) \quad (11)$$

$$Lab' = (L, a * (1 + \Delta C), b * (1 + \Delta C)) \quad (12)$$

$$Lab' = (L, a * \cos(\Delta H) - b * \sin(\Delta H), a * \sin(\Delta H) + b * \cos(\Delta H)) \quad (13)$$

To make the color adjustments happen at specific angles, the aspecular angle,  $\theta_{crosshair}$ , of the position under the crosshair is used to compute the ratio,  $R$ , of how the face and flop color will be adjusted. Equation 14 maps  $\theta_{crosshair}$  from between  $15^\circ$  and the  $\theta_{clamp}$  angle to 0 to 1. If the angle is less than  $15^\circ$  it treats it as the face and assigns the value of 1.0 to  $R$ . If it is greater than the color's  $\theta_{clamp}$  it treats it as the flop and sets  $R$  to 0.0.

$$R = \begin{cases} 0 & 15^\circ > \theta_{crosshair} \\ \frac{\theta_{crosshair} - 15^\circ}{\theta_{clamp} - 15^\circ} & 15^\circ \geq \theta_{crosshair} \geq \theta_{clamp} \\ 1 & \theta_{crosshair} > \theta_{clamp} \end{cases} \quad (14)$$

The temporary colors are blended with the original color by the ratio,  $R$ , to get the final colors, marked by a double prime symbol, for face and flop:

$$Lab''(15^\circ) = (1 - R) * Lab(15^\circ) + R * Lab'(15^\circ) \quad (15)$$

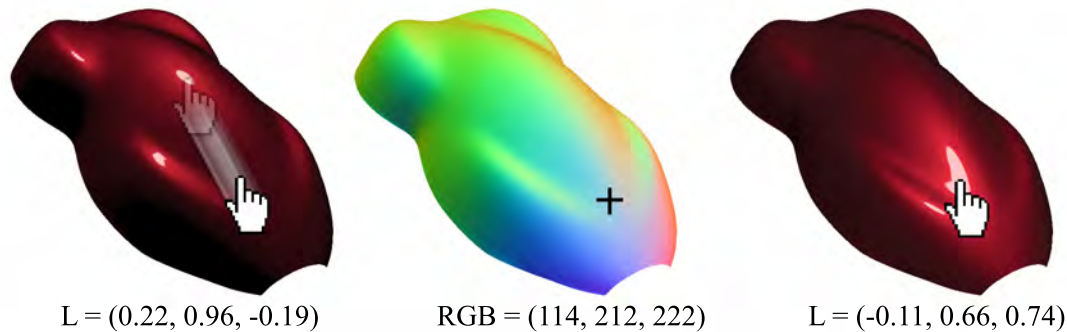
$$Lab''(\theta_{clamp}) = R * Lab(\theta_{clamp}) + (1 - R) * Lab'(\theta_{clamp}) \quad (16)$$

### Adjust travel

When the mouse is depressed, the system samples the off screen buffer to retrieve the aspecular angle at the coordinates under the cursor,  $\theta_{cursor}$ . In the first scenario the user clicked on the ring visualizing travel. This is detected by comparing  $\theta_{cursor}$  to  $\theta_{clamp}$  to determine if it is within a few degrees. If the user wants to adjust travel to coincide with the curvature of the object, they can click and drag the ring representing  $\theta_{clamp}$ . They set the position where the color travel visually stops by using the mouse to move the spot directly on the object. As the user drags the mouse, the  $\theta_{clamp}$  (the travel parameter of the face flop model) is set based on the aspecular angle under the new position of the cursor.

### Adjust gloss

Adjusting the size of the gloss lobe works in a manner similar to adjusting travel. When the user clicks and drags the edge of the gloss, the gloss level is adjusted based on the new aspecular angle under the cursor as it repositioned. A mapping between gloss and aspecular angles needs to be made to detect when the edge of gloss is being clicked on. The implementation is relatively straightforward.



**Figure 9.** The stylist begins to move the light by clicking one the specular highlight and dragging it to the desired location (left). The reflected view vector in direction the light needs to be in to produce a specular highlight. A shader encodes the reflected view vector as RGB triplets into an off screen buffer (middle). The light position is set to be the vector encoded in the buffer at the mouse cursor coordinates (right).

### Move light

Finally, we describe a simple but useful tool for positioning the light by dragging the specular highlight illustrated in Figure 9. To determine if the user is trying to click on the highlight, the system samples the aspecular angle of the mouse cursor position from the off screen buffer,  $\theta_{cursor}$ . The gloss parameter determines the shape and size of the specular lobe of the BRDF. By sampling the gloss lobe with the aspecular angle of the cursor position one can determine if the user has clicked on the center of the highlight (to reposition the light), has clicked the edge of the highlight (to adjust the gloss level), or clicking the object to rotate it.

When the user clicks near the center of a specular highlight and begins to drag it, a different shader is used for the off screen buffer. Instead of computing the aspecular angle, this shader computes at every pixel on the object the direction that the light would need to be in if the specular highlight was centered on that particular pixel. The required light vector is simply the view vector reflected off the surface normal. This vector is stored in the off screen buffer encoded as RGB values as shown in Figure 9 center. As the user drags the cursor across the object, the system sets the new position of the light to be the light vector that is encoded in the off screen buffer at the coordinates of the cursor. The light is updated and specular highlight tracks the mouse cursor.

### Evaluation

We evaluated the color styling interface by asking several designers familiar with automotive metallic surface coatings to use our software to create several new colors. We began our informal tests by using real paint samples to make sure that each designer understood what was meant by the terms face color, flop color, travel, and gloss. This evaluation was accomplished by asking each participant to correctly orient the sample with respect to the light source so that they could view face color, the flop color, or some color that was part of the color travel between these two extremes. The participants were then requested to perform a color styling task using both the interface described in this paper and the goniochromatic color design interface shown in Figure 3. The designers were asked to provide their reactions to using the new and the old interfaces.

**Select the face color, flop color, travel, and gloss for a new metallic color:** In our new interface this is accomplished by using the virtual mood board to choose the face color, the flop color, and the gloss for the color. The image based BRDF tweaker is employed to determine the travel and adjust the color. In the existing goniochromatic color design interface this task is accomplished by designating the face and flop colors on a hue/saturation diagram, adjusting the L curve for the color, and setting the position of a gloss slider (see Figure 3).

The participants remarked that the pictures on the mood board interface suggested better starting points for their color design and allowed them to use their own reference material to seed the design process. They found it difficult to find a particular color using the single color wheel available in the existing goniochromatic color design interface. Using the BRDF tweaker to point at the surface and adjust the color travel was considered to be more straightforward than using the old goniochromatic color design interface to manipulate the endpoints of a second order curve.

**Adjust the hue, saturation, and brightness of an existing metallic color:** This is done with the new color styling interface by moving the color adjustment tool in the BRDF tweaker (see Figure 7) across the surface of the object and making a change to the color (lighten/darken, saturate/desaturate, etc.) at that location. In the old goniochromatic color design interface (see Figure 3) this is done indirectly by using a hue, saturation, and brightness interface located next to the window containing the object.

The participants said that using the BRDF tweaker was more straightforward because they could make adjustments directly on the surface of the object. This made it clear where the changes would take effect. It was also unnecessary to identify the color as face or flop because the position of the tool accomplished this implicitly. Adjusting the travel of the color directly on the surface made it possible to create a color that accentuates the attractive geometric properties of the object (or de-emphasizes the un-desirable geometric attributes).

### Conclusions

We have proposed a simple but powerful reflection model that can accurately represent the solid and metallic colors that are of practical importance in most traditional design applications.

The model includes a straightforward curve drawing interface that can be used to directly set the model's limited number of parameters, and, because of its simplicity, the model allows indirect selection of its variables via design oriented interfaces. The parametric form of the face, flop, travel, and gloss reflection model also makes it easy to use in rendering applications.

This paper has also suggested two novel color styling interfaces that take advantage of the underlying simplicity of the proposed face, flop, travel, and gloss reflection model. These interfaces attempt to mimic the typical work flows that are found in the fashion, interior, and industrial design fields. The virtual mood board allows designers to use exemplars to directly inspire new color appearances. Pictures of collected color artifacts can be organized and displayed, and values for the parameters of the reflection model can be selected by making direct reference to these illustrations. The spot color styler interface allows changes to color and appearances to be made directly on a 3D object. This approach allows the shape of the object and the lighting to be considered as the color is manipulated.

## References

- [1] David H. Alman. Directional color measurement of metallic flake finishes. ISCC Williamsburg Conf. on Appearance, 53–56, 1987.
- [2] Michael Ashby and Kara Johnson. *Materials and Design: The Art and Science of Material Selection in Product Design*, 41–42. Butterworth-Heinemann, 2002.
- [3] Mark Colbert, Sumanta Pattanaik, and Jaroslav Krivanek. Brdf-shop: Creating physically correct bidirectional reflectance distribution functions. *IEEE Comput. Graph. Appl.*, 26(1) : 30–36, 2006.
- [4] Maria Lurdes Dias. Ray tracing interference color. *IEEE Computer Graph. Appl.*, 11(2) : 54–60, 1991.
- [5] Sergey Ershov, Konstantin Kolchin, and Karol Myszkowski. Rendering pearlescent appearance based on paint-composition modelling. *Proc. Eurographics '01*, 20(3) : 227–238, 2001.
- [6] Johannes Günther, Tongbo Chen, Michael Goesele, Ingo Wald, and Hans-Peter Seidel. Efficient acquisition and realistic rendering of car paint. *Vision, Modeling, and Visualization '05*, 487–494, 2005.
- [7] Hideki Hirayama, Kazufumi Kaneda, Hideo Yamashita, Yoshiki Yamaji, and Yoshimi Monden. Visualization of optical phenomena caused by multilayer films based on wave optics. *The Visual Computer*, 17(2) : 106–120, 2001.
- [8] Jan Kautz. *Game Programming Gems 3, Rendering with Handcrafted Shading Models*, 477–485. Charles River, 2002.
- [9] Erum Arif Khan, Erik Reinhard, Roland W. Fleming, and Heinrich H. Bühlhoff. Image-based material editing. *ACM Trans. Graph.*, 25(3) : 654–663, 2006.
- [10] Hendrik P. A. Lensch, Michael Goesele, Jan Kautz, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. *Eurographics Workshop on Rendering Techniques*, 103–114, 2001.
- [11] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. *Computer Graphics*, 31(Annual Conference Series) : 389–400, 1997.
- [12] Addy Ngan, Fredo Durand, and Wojciech Matusik. Image-driven navigation of analytical brdf models. *Eurographics Rendering Workshop '06*, 399–408, 2006.
- [13] Fabio Pellacini, James A. Ferwerda, and Donald P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. *Proc. SIGGRAPH '00*, 55–64, 2000.
- [14] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. *Proc. SIGGRAPH '01*, 117–128, 2001.
- [15] Allan B. J. Rodrigues. Measurement of metallic and pearlescent finishes. *Die Farbe*, 37 : 65–78, 1990.
- [16] Martin Rump, Gero Müller, Ralf Sarlette, Dirk Koch, and Reinhard Klein. Photo-realistic rendering of metallic car paint from image-based measurements. *Computer Graphics Forum*, 27(2), 2008.
- [17] Clement Shimizu and Gary W. Meyer. Computer aided color appearance design using environment map based lighting. *Eurographics Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, 223–230, 2005.
- [18] Clement Shimizu, Gary W. Meyer, and Joseph P. Wingard. Interactive goniochromatic color design. *11th Color Imaging Conference*, 16–22, 2003.
- [19] Peter-Pike J. Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere: a model for capturing npr shading from art. *Graphics interface*, 143–150, 2001.
- [20] Roman Āurikovič and Tomaš Āgošton. Prediction of optical properties of paints. *Central European Journal of Physics*, 5(3) : 416–427.
- [21] William H. Venable. A model for interpreting three-angle measurements of flake finishes. *ISCC Williamsburg Conf. on Appearance*, 57–60, 1987.
- [22] Harold B. Westlund and Gary W. Meyer. Applying appearance standards to light reflection models. *SIGGRAPH '01*, 501–510, 2001.
- [23] Tsung-Yi Wu, Wan-Chun Ma, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung. Image-based brdf acquisition for non-spherical objects. *Workshop on Computer Vision & Graphic Image Processing '05*, 2005.
- [24] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: recovering reflectance models of real scenes from photographs. *Proc. SIGGRAPH '99*, 215–224, 1999.

## Acknowledgements

The authors would like to thank Kevin Ellwood, John Hall, and Mark Nichols from Ford Motor Company for their help in understanding how color paint is selected and controlled in the automotive manufacturing process. Allan Rodriguez and Dave Griffus of DuPont helped us turn color specifications into paint formulations. Molly Roark and Kate Iverson provided artistic support for the mood boards, and Stephen Stephens assisted in videotaping our work. This research was supported by a grant from the Ford Motor Company.

## Author Biography

*Clement Shimizu received his BS in Math and is defending his PhD. in Computer Science at the University of Minnesota. He has dedicated his life to serving artists, designers, and other creative people through technological innovation. He does this through his academic research in computer graphics, his work in immersive projection systems for the Elumenati, and in his personal life volunteering engineering services to the vibrant arts and design community.*

*Gary Meyer is an Associate Professor in the Department of Computer Science and Engineering at the University of Minnesota. He has also been a member of the Computer Science faculty at the University of Oregon and a Member of Technical Staff at Bell Telephone Laboratories. Meyer received his Bachelors Degree from the University of Michigan, his Masters Degree from Stanford University, and his PhD Degree from Cornell University.*