Chapter 12

Color Models and Color Applications

Color concepts and programming applications for color models are explored in the exercises for this chapter.

Exercises

12-1. A color position in RGB space is specified with parameters that are all in the range from 0.0 to 1.0, and the corresponding HSV parameters can also be normalized to values in the range from 0.0 to 1.0. Parameter V is assigned the value 0.0 at the apex of the hexcone (the black point) and the value 1.0 at the hexagon base. Parameter S is assigned the value 0.0 for any gray-scale color (along the axis of the hexcone) and the value 1.0 for all color points along the sides of the hexcone, including the perimeter of the base. And parameter H can be assigned the value 0.0 at the red color point on the base of the hexcone, corresponding to an angle of 0°, and the value 1.0 at the red color point, representing a 360° angle around the V axis.

Any point on the surface of the RGB color cube, where $max(R,\, G,\, B) = 1.0$, is mapped to a position on the hexagon base of the HSV hexcone, where $V = 1.0$. Similarly, any point within the RGB color cube is on the surface of a subcube whose edge length is equal to the value of the maximum RGB component for that color point. This subcube transforms into a subhexcone of the HSV space, with its apex at the black point, and the RGB color point is thus mapped to a position on the base of that subhexcone. Therefore, in HSV space, the value of parameter V for any RGB color point is equal to the value of $max(R,\, G,\, B)$.

The value of parameter S (saturation) for a color point represents the relative distance of the point from the axis of the hexcone. Along the hexcone axis, $min(R,\, G,\, B) = max(R,\, G,\, B)$ and $S = 0.0$ (gray-scale positions), and at any position along the sides of the hexcone, $S = 1.0$. Thus,

$$S = \begin{cases} 0.0, & \text{if } max(R,\ G,\ B) = 0 \\ \frac{max(R,\ G,\ B) - min(R,\ G,\ B)}{max(R,\ G,\ B)}, & \text{if } max(R,\ G,\ B) \neq 0 \end{cases}$$

Parameter H is a measure of the angular displacement of a color point around the V axis, measured counterclockwise from the S axis, when the origin of HSV space is viewed from a position on the positive V axis. For any gray-scale color ($S = 0.0$), the value of H is undefined. Otherwise, the value for H can be determined by calculating the offset of the color point from the centerline of the hexagon trisection on the subhexcone base that contains that point. The centerlines for the hexagon trisections on the base of the HSV hexcone are the lines from the white point to the red point, the green point, and the blue point.

Any color point that is on the "red" face of the RGB unit cube (whose vertices are the red, magenta, white, and yellow color points) has a value of 1.0 for the red component of that color. This face of the RGB cube maps to the hexagon trisection on the hexcone base that has an angular spread of $\pm 60°$ relative to the trisection centerline, which is the line from the white point to the red point (parallel to the S axis). If the other two RGB components for the color point are $G = 1.0$ and $B = 0.0$, then $H = 60°$ (the yellow point). If $G = 0.0$ and $B = 1.0$, then $H = 300°$ (the magenta point). And if $G = B$, then $H = 0°$ (a red "tint" position on the trisection centerline).

A similar mapping holds for any color point within the RGB unit cube that is on the "red" face of a subcube. The red component for this color has a value that is equal to the maximum of the RGB color components, and the RGB color point maps to a position on the base of a subhexcone that is within the hexagon trisection that is centered about the S axis. Thus, for any color point whose red value is equal to the maximum of the RGB components, the relative offset value for H (in the range from -1.0 to 1.0) is calculated as

$$H = \frac{G\ -\ B}{max(R,\ G,\ B)\ -\ min(R,\ G,\ B)}$$

This normalized value is then multiplied by 60 to obtain an angular measure within the hexagon trisection. Also, 360 is added to this value if H is negative, so that all angular values for this hexcone trisection are positive and within the interval from $300°$ ($-60°$) to $60°$.

A color point that is on the "green" face of an RGB subcube maps to a position in HSV space such that parameter H has a value in the range from $60°$ to $180°$. The green component for this color has a value that is equal to the maximum of the RGB color components. Offset calculations from the centerline of this hexagon trisection are similar to those for the "red" section, except that we also need to add 2.0 to the offset so that it is shifted $120°$ when multiplied by 60:

$$H = 2.0 + \frac{B\ -\ R}{max(R,\ G,\ B)\ -\ min(R,\ G,\ B)}$$

A color point that is on the "blue" face of an RGB subcube maps to a position in HSV space such that parameter H has a value in the range from 180° to 300°. The blue component for this color has a value that is equal to the maximum of the RGB color components. Offset calculations from the centerline of this hexagon trisection are similar to those for the "red" section, except that we also need to add 4.0 to the offset so that it is shifted 240° when multiplied by 60:

$$H = 4.0 + \frac{R \,-\, G}{max(R,\, G,\, B) \,-\, min(R,\, G,\, B)}$$

An angular value for parameter H can be normalized so that $0.0 \le H < 1.0$ by dividing the value by 360. This represents the range of angular values around the V axis, starting at the red color point, with $H = 0.0$ corresponding to both 0° and 360°.

12-2. All input HSV parameters can be specified in the range form 0.0 to 1.0, or the H parameter could be an angular value in the range $0° \le H < 360°$. Given a set of input HSV color parameters, we can first check parameter S to determine whether the color point is on the gray-scale line. If $S = 0.0$, then $R = G = B$, and the value for all three RGB components is the same as the value of parameter V.

Otherwise, check parameter H to determine the angular interval within HSV space that contains the input color point. This interval identifies the maximum RGB component. For example, if $300° \le H < 60°$, then $R = V$ because V is equal the maximum of the RGB parameters for any color point and the "red" face of an RGB subcube maps to this trisection on the base of an HSV subhexcone (Exercise 12-1). Similarly, if $60° \le H < 120°$, then $G = V$ because the "green" face of an RGB subcube maps to this trisection on the base of an HSV subhexcone. And if $180° \le H < 300°$, then $B = V$ because the "blue" face of an RGB subcube maps to this trisection on the base of an HSV subhexcone.

Next, identify the minimum RGB component using the angular interval within HSV space and the expression for parameter S, which is

$$S = \frac{V \,-\, min(R,\, G,\, B)}{V},$$

Therefore,

$$min(R,\, G,\, B) = V(1.0 \,-\, S)$$

And, from Exercise 12-1,

$$min(R,\ G,\ B) = \begin{cases} B, & \text{if } 0^\circ \le H < 60^\circ \\ B, & \text{if } 60^\circ \le H < 120^\circ \\ R, & \text{if } 120^\circ \le H < 180^\circ \\ R, & \text{if } 180^\circ \le H < 240^\circ \\ G, & \text{if } 240^\circ \le H < 300^\circ \\ G, & \text{if } 300^\circ \le H < 360^\circ \end{cases}$$

The value for the remaining RGB component can then be obtained from the expressions for H and S. For example, on the "red" face of an RGB subcube, the value of parameter H for any color point is calculated as (Exercise 12-1)

$$H = \frac{G - B}{V - min(R,\ G,\ B)}$$

In the angular interval from 0° to 60° within HSV space, the minimum RGB component is parameter B, so that

$$H = \frac{G - V(1.0 - S)}{V - V(1.0 - S)} = \frac{G - V(1.0 - S)}{V \cdot S}$$

Therefore,

$$\begin{aligned} G &= H \cdot V \cdot S + V(1.0 - S) \\ &= V[1.0 - S(1.0 - H)] \end{aligned}$$

But in the angular interval from 300° to 0°, the minimum RGB component is G, and solving the equation $H = (G - B)(V - min(R,\ G,\ B))$ for parameter B produces a negative value, which represents the fractional displacement for a negative angle relative to the S axis. Thus, to calculate the correct value for parameter B (in the interval from 0.0 to 1.0), we first need to convert H to a value that represents the positive fractional distance across the sextant from the left boundary line, which is at an angle of −60° relative to the S axis. This converted calculation is

$$H = 1.0 - \frac{B - G}{V - min(R,\ G,\ B)}$$

or

$$H = 1.0 - \frac{B - V(1.0 - S)}{V \cdot S}$$

and the value for parameter B is calculated as

$$B = V(1.0 - S \cdot H)$$

Similar calculations are performed in the remaining two trisections on the base of an HSV subhexcone, which correspond to the "green" and "blue" faces of an RGB subcube.

An efficient algorithm for calculating RGB components can be devised using an input normalized value for H in the interval from 0.0 to 1.0. Multiplying H by 6 converts the input value to the range from $0.0 \le H < 6.0$, and $floor(H)$ can be used to label each of the six angular regions (sextants) on the base of any HSV subhexcone. For example, the "red" face of an RGB subcube maps to the trisection on the base of an HSV subhexcone that contains the two sextants labeled 5 and 0 (that is, $5.0 \le H < 5.99$ and $0.0 \le H < 0.99$), and $R = V = max(R, G, B)$ in these two sextants. Also, $G = V$ in sextants 1 and 2, $B = V$ in sextants 3 and 4. The minimum RGB value is B in sextants 0 and 1, R in sextants 2 and 3, and G in sextants 4 and 5.

The calculations for the RGB components in the six sextants can be performed using the following parameters

$$\begin{aligned} fracH &= H - floor(H) \\ minRGB &= V(1.0 - S) \\ posSxtnt &= V[1.0 - S(1.0 - H)] \\ negSxtnt &= V(1.0 - S \cdot H) \end{aligned}$$

And the RGB components in the sextants are

$$(R,\ G,\ B) = \begin{cases} (V,\ posSxtnt,\ minRGB), & \text{in sextant 0} \\ (negSxtnt,\ V,\ minRGB), & \text{in sextant 1} \\ (minRGB,\ V,\ posSxtnt), & \text{in sextant 2} \\ (minRGB,\ negSxtnt,\ V), & \text{in sextant 3} \\ (posSxtnt,\ minRGB,\ V), & \text{in sextant 4} \\ (V,\ minRGB,\ negSxtnt), & \text{in sextant 5} \end{cases}$$

12-3. Methods from Section 11-7 can be used to develop a menu program that allows selection of the HSV components using a mouse. Or sliders or a color wheel could be displayed to allow interactive color selection on a numerical scale, and interpolation calculations can be performed to calculate values on the scale. The selected set of HSV color values are then converted to RGB values using the algorithm discussed in Exercise 12-2.

12-4. The procedures discussed for Exercise 11-5 can be employed to develop the slider displays for this program. Values for parameters H and S are selected in the range from 0.0 to 1.0, and selected H values could be in the range from $0°$ to $360°$. The selected numerical values can be displayed for user verification within a small rectangle placed next to each slider.

12-5. In the program of the preceding exercise, convert the selected HSV values into RGB values (see Exercise 12-2), and display these values as well as the selected HSV values.

12-6. The previous program is to be modified so that a small color rectangle is displayed next to the selected RGB echo values, where the displayed color is determined by the selected color parameters.

12-7. The HLS color space can be represented as either a double cone or a double hexcone. As in HSV space, the color black is represented by the point at $L = 0.0$ and $S = 0.0$, the color white is represented by the point at $L = 1.0$ and $S = 0.0$, and, at any position along the gray-scale line (the axis of the double cone or double hexcone), $S = 0.0$ and H is undefined. But in the HLS model, the spectral colors are all represented as positions on the $L = 0.5$ plane, with $S = 1.0$ and parameter H varying from $0°$ to $360°$. Any spectral color could be selected as the reference position for parameter H, such as the blue point in Fig. 12-18.

Input RGB values are specified in the range from 0.0 to 1.0, and calculations similar to those in Exercise 12-1 are used to obtain values for the HLS parameters. Any point on the surface of the RGB color cube, where $max(R, G, B) = 1.0$, is mapped to a position on the base of the HLS double cone (or double hexcone), where $L = 0.5$. And any spectral color point in RGB space has $max(R, G, B) = 1.0$ and $min(R, G, B) = 0.0$. Also, the white point parameter values are $R = G = B = 1.0$, and the black point has the color components $R = G = B = 0.0$. Therefore

$$L = \frac{max(R,\ G,\ B)\ +\ min(R,\ G,\ B)}{2}$$

As in the HSV model, parameter S (saturation) is the relative distance from the axis of the HLS double cone (or double hexcone). However, the distance from this axis to the boundary surface of HLS space first increases, as L varies from 0.0 to 0.5, then decreases, as L varies from 0.5 to 1.0. Thus, the value of parameter S is

$$S = 0.0, \qquad \text{if } max(R,\ G,\ B) = 0$$

otherwise

$$S = \frac{max(R,\ G,\ B)\ -\ min(R,\ G,\ B)}{max(R,\ G,\ B)\ +\ min(R,\ G,\ B)}, \qquad \text{if } L \le 0.5$$

or

$$S = \frac{max(R,\ G,\ B)\ -\ min(R,\ G,\ B)}{2.0\ -\ (max(R,\ G,\ B)\ +\ min(R,\ G,\ B))}, \qquad \text{if } L > 0.5$$

Calculations for parameter H in HLS space are the same as in HSV space (Exercise 12-1), except the $H = 0°$ position could be taken as the blue point: $(R, G, B) = (0.0, 0.0, 1.0)$.

12-8. All input HLS parameters can be specified in the range form 0.0 to 1.0, or the H parameter could be an angular value in the range $0° \le H < 360°$. Given a set of input

HLS color parameters, we can first check parameter S to determine whether the color point is on the gray-scale line. If $S = 0.0$, then $R = G = B$. And, from Exercise 12-7,

$$L = \frac{max(R, G, B) + min(R, G, B)}{2}$$

so that the value for all three RGB components of any gray-scale color is the same as the value of parameter L.

Otherwise, check parameter H to determine the angular interval within HLS space that contains the input color point. This interval identifies the maximum RGB component. For example, if we take $H = 0°$ at the spectral red position, then in the interval $300° \leq H < 60°$, R is the maximum component because the "red" face of the RGB color cube maps to this region (Exercise 12-1). And we can obtain the value for parameter R in this region from the expression for evaluating parameter L:

$$R = 2\,L - min(R, G, B)$$

Similarly, if $60° \leq H < 180°$, then the maximum RGB component is G. And if $180° \leq H < 300°$, the maximum RGB component is B. (Alternatively, we could set $H = 0°$ at any other reference, such as the blue point, as in Fig. 12-18.)

Once the maximum RGB component has been identified, the expression for calculating parameter S can be used to obtain an expression for the minimum RGB component. For example, when $L \leq 0.5$

$$S = \frac{max(R,\ G,\ B) - min(R,\ G,\ B)}{max(R,\ G,\ B) + min(R,\ G,\ B)}$$

and

$$min(R,\ G,\ B) = \frac{(1 - S)\,max(R,\ G,\ B)}{1 + S}$$

Then the appropriate expression for H is used to obtain the final RGB component.

12-9. Input for this program is the set of seven numerical values specifying two RGB color points, (R_0, G_0, B_0) and $(R_{end}, G_{end}, B_{end})$, and the number of colors to be selected. The input could be specified in a data file or with typed values from a keyboard. Alternatively, the two input colors could be interactively selected using sliders or a color wheel.

Linear interpolation is used to compute values between the two input color points, and interactive slider selection (Exercise 12-4) can be used to obtain the selected colors. The slider is used to select a parameter u in the range from 0.0 to 1.0, then each color component is calculated using the selected parameter value. For example, the value for the red component is calculated as

$$R = R_0 + (R_{end} - R_0)\,u$$

and similar calculations are used for the green and blue components.

12-10. This program is a modification of the procedures discussed for the previous exercise. Instead of linearly interpolating color values, each *RGB* component could be selected using a displayed slider, where the initial and final numerical values for each slider is determined from the input. This allows any color point to be selected from within a specified three-dimensional subregion of the *RGB* color cube.

12-11. This program is a slight modification of the Exercise 12-9 program. All color parameters could be selected in the range from 0.0 to 1.0, or the numerical value for the H parameter could be selected in the range from $0°$ to $360°$. The selected *HSV* parameters could be converted to *RGB* values and displayed in echo boxes.

12-12. This program is also a slight modification of the Exercise 12-9 program. All color parameters could be selected in the range from 0.0 to 1.0, or the numerical value for the H parameter could be selected in the range from $0°$ to $360°$. The selected *HLS* parameters could be converted to *RGB* values and displayed in echo boxes.

12-13. For one rectangle, use a random-number function to generate three values, each in the range from 0.0 to 1.0, for each pixel position within the rectangle. These three random numbers are then used as values for the *RGB* components at that position, and the pixel color is set with the OpenGL color command. For the second rectangle, generate one random number, in the range from 0.0 to 1.0, for each pixel position and use that single value to select a set of *RGB* components from a subregion of the *RGB* color cube, using the methods discussed for Exercise 12-10. Compare the two color rectangles for a number of different *RGB* subspaces and summarize your results.

12-14. This exercise is a modification of the preceding exercise, where color selections are determined from either the *HSV* or *HLS* color space. Input color parameters are to be converted to *RGB* values, and pixels are displayed using the OpenGL color command.