7.2.4 Blend Mode

In principle, the blend function $B(C_b, C_s)$, used in the compositing formula to customize the blending operation, could be any function of the backdrop and source colors that yields another color, C_r, for the result. PDF defines a standard set of named blend functions, or *blend modes*, listed in Tables 7.2 and 7.3. Plates 18 and 19 illustrate the resulting visual effects for *RGB* and *CMYK* colors, respectively.

A blend mode is termed *separable* if each component of the result color is completely determined by the corresponding components of the constituent backdrop and source colors—that is, if the blend mode function B is applied separately to each set of corresponding components:

$$c_r = B(c_b, c_s)$$

where the lowercase variables c_r, c_b, and c_s denote corresponding components of the colors C_r, C_b, and C_s, expressed in additive form. (Theoretically, a blend mode could have a different function for each color component and still be separable; however, none of the standard PDF blend modes have this property.) A separable blend mode can be used with any color space, since it applies independently to any number of components. Only separable blend modes can be used for blending spot colors.

Table 7.2 lists the standard separable blend modes available in PDF.

TABLE 7.2 Standard separable blend modes

NAME	RESULT
Normal	Selects the source color, ignoring the backdrop: $B(c_b, c_s) = c_s$
Multiply	Multiplies the backdrop and source color values: $B(c_b, c_s) = c_b \times c_s$ The result color is always at least as dark as either of the two constituent colors. Multiplying any color with black produces black; multiplying with white leaves the original color unchanged. Painting successive overlapping objects with a color other than black or white produces progressively darker colors.

NAME	RESULT
Screen	Multiplies the complements of the backdrop and source color values, then complements the result: $B(c_b, c_s) = 1 - [(1 - c_b) \times (1 - c_s)] = c_b + c_s - (c_b \times c_s)$ The result color is always at least as light as either of the two constituent colors. Screening any color with white produces white; screening with black leaves the original color unchanged. The effect is similar to projecting multiple photographic slides simultaneously onto a single screen.
Overlay	Multiplies or screens the colors, depending on the backdrop color value. Source colors overlay the backdrop while preserving its highlights and shadows. The backdrop color is not replaced but is mixed with the source color to reflect the lightness or darkness of the backdrop. $B(c_b, c_s) = \text{HardLight}(c_s, c_b)$
Darken	Selects the darker of the backdrop and source colors: $B(c_b, c_s) = \min(c_b, c_s)$ The backdrop is replaced with the source where the source is darker; otherwise, it is left unchanged.
Lighten	Selects the lighter of the backdrop and source colors: $B(c_b, c_s) = \max(c_b, c_s)$ The backdrop is replaced with the source where the source is lighter; otherwise, it is left unchanged.
ColorDodge	Brightens the backdrop color to reflect the source color. Painting with black produces no changes. $B(c_b, c_s) = \begin{cases} \min(1, c_b / (1 - c_s)) & \text{if } c_s < 1 \\ 1 & \text{if } c_s = 1 \end{cases}$
ColorBurn	Darkens the backdrop color to reflect the source color. Painting with white produces no change. $B(c_b, c_s) = \begin{cases} 1 - \min(1, (1 - c_b) / c_s) & \text{if } c_s > 0 \\ 0 & \text{if } c_s = 0 \end{cases}$

NAME	RESULT
HardLight	Multiplies or screens the colors, depending on the source color value. The effect is similar to shining a harsh spotlight on the backdrop. $$B(c_b, c_s) = \begin{cases} \text{Multiply}(c_b, 2 \times c_s) & \text{if } c_s \le 0.5 \\ \text{Screen}(c_b, 2 \times c_s - 1) & \text{if } c_s > 0.5 \end{cases}$$
SoftLight	Darkens or lightens the colors, depending on the source color value. The effect is similar to shining a diffused spotlight on the backdrop. $$B(c_b, c_s) = \begin{cases} c_b - (1 - 2 \times c_s) \times c_b \times (1 - c_b) & \text{if } c_s \le 0.5 \\ c_b + (2 \times c_s - 1) \times (\text{D}(c_b) - c_b) & \text{if } c_s > 0.5 \end{cases}$$ where $$\text{D}(x) = \begin{cases} ((16 \times x - 12) \times x + 4) \times x & \text{if } x \le 0.25 \\ \sqrt{x} & \text{if } x > 0.25 \end{cases}$$
Difference	Subtracts the darker of the two constituent colors from the lighter color: $$B(c_b, c_s) = \lvert c_b - c_s \rvert$$ Painting with white inverts the backdrop color; painting with black produces no change.
Exclusion	Produces an effect similar to that of the **Difference** mode but lower in contrast. Painting with white inverts the backdrop color; painting with black produces no change. $$B(c_b, c_s) = c_b + c_s - 2 \times c_b \times c_s$$

Table 7.3 lists the standard nonseparable blend modes. Since the nonseparable blend modes consider all color components in combination, their computation depends on the blending color space in which the components are interpreted. They may be applied to all multiple-component color spaces that are allowed as blending color spaces (see Section 7.2.3, “Blending Color Space”).

All of these blend modes conceptually entail the following steps:

1. Convert the backdrop and source colors from the blending color space to an intermediate *HSL* (hue-saturation-luminosity) representation.
2. Create a new color from some combination of hue, saturation, and luminosity components selected from the backdrop and source colors.

3. Convert the result back to the original (blending) color space.

However, the formulas given below do not actually perform these conversions. Instead, they start with whichever color (backdrop or source) is providing the hue for the result; then they adjust this color to have the proper saturation and luminosity.

The nonseparable blend mode formulas make use of several auxiliary functions. These functions operate on colors that are assumed to have red, green, and blue components. (Blending of *CMYK* color spaces requires special treatment, as described below.)

$$\text{Lum}(C) = 0.3 \times C_{\text{red}} + 0.59 \times C_{\text{green}} + 0.11 \times C_{\text{blue}}$$

$$\begin{aligned}
&\text{SetLum}(C, l) \\
&\qquad \text{let } d = l - \text{Lum}(C) \\
&\qquad C_{\text{red}} = C_{\text{red}} + d \\
&\qquad C_{\text{green}} = C_{\text{green}} + d \\
&\qquad C_{\text{blue}} = C_{\text{blue}} + d \\
&\qquad \text{return ClipColor}(C)
\end{aligned}$$

$$\begin{aligned}
&\text{ClipColor}(C) \\
&\qquad \text{let } l = \text{Lum}(C) \\
&\qquad \text{let } n = \min(C_{\text{red}}, C_{\text{green}}, C_{\text{blue}}) \\
&\qquad \text{let } x = \max(C_{\text{red}}, C_{\text{green}}, C_{\text{blue}}) \\
&\qquad \text{if } n < 0.0 \\
&\qquad\quad C_{\text{red}} = l + (((C_{\text{red}} - l) \times l) / (l - n)) \\
&\qquad\quad C_{\text{green}} = l + (((C_{\text{green}} - l) \times l) / (l - n)) \\
&\qquad\quad C_{\text{blue}} = l + (((C_{\text{blue}} - l) \times l) / (l - n)) \\
&\qquad \text{if } x > 1.0 \\
&\qquad\quad C_{\text{red}} = l + (((C_{\text{red}} - l) \times (1 - l)) / (x - l)) \\
&\qquad\quad C_{\text{green}} = l + (((C_{\text{green}} - l) \times (1 - l)) / (x - l)) \\
&\qquad\quad C_{\text{blue}} = l + (((C_{\text{blue}} - l) \times (1 - l)) / (x - l)) \\
&\qquad \text{return } C
\end{aligned}$$

$$\text{Sat}(C) = \max(C_{\text{red}}, C_{\text{green}}, C_{\text{blue}}) - \min(C_{\text{red}}, C_{\text{green}}, C_{\text{blue}})$$

In the following function, the subscripts *min*, *mid*, and *max* refer to the color components having the minimum, middle, and maximum values upon entry to the function.

SetSat(C, s)
  if $C_{max} > C_{min}$
    $C_{mid} = (((C_{mid} - C_{min}) \times s) / (C_{max} - C_{min}))$
    $C_{max} = s$
  else
    $C_{mid} = C_{max} = 0.0$
  $C_{min} = 0.0$
  return C

TABLE 7.3 Standard nonseparable blend modes

NAME	RESULT
Hue	Creates a color with the hue of the source color and the saturation and luminosity of the backdrop color. $B(C_b, C_s) = \text{SetLum}(\text{SetSat}(C_s, \text{Sat}(C_b)), \text{Lum}(C_b))$
Saturation	Creates a color with the saturation of the source color and the hue and luminosity of the backdrop color. Painting with this mode in an area of the backdrop that is a pure gray (no saturation) produces no change. $B(C_b, C_s) = \text{SetLum}(\text{SetSat}(C_b, \text{Sat}(C_s)), \text{Lum}(C_b))$
Color	Creates a color with the hue and saturation of the source color and the luminosity of the backdrop color. This preserves the gray levels of the backdrop and is useful for coloring monochrome images or tinting color images. $B(C_b, C_s) = \text{SetLum}(C_s, \text{Lum}(C_b))$
Luminosity	Creates a color with the luminosity of the source color and the hue and saturation of the backdrop color. This produces an inverse effect to that of the **Color** mode. $B(C_b, C_s) = \text{SetLum}(C_b, \text{Lum}(C_s))$

The above formulas apply to *RGB* spaces. Blending in *CMYK* spaces (including both **DeviceCMYK** and **ICCBased** calibrated *CMYK* spaces) is handled in the following way:

- The *C*, *M*, and *Y* components are converted to their complementary *R*, *G*, and *B* components in the usual way. The formulas above are applied to the *RGB* color values. The results are converted back to *C*, *M*, and *Y*.
- For the *K* component, the result is the *K* component of C_b for the **Hue**, **Saturation**, and **Color** blend modes; it is the *K* component of C_s for the **Luminosity** blend mode.

Note: *An additional standard blend mode,* ***Compatible****, is a vestige of an earlier design and is no longer needed but is still recognized for the sake of compatibility. Its effect is equivalent to that of the* ***Normal*** *blend mode. See "Compatibility with Opaque Overprinting" on page 567 for further discussion.*

7.2.5 Interpretation of Alpha

The color compositing formula

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) \times C_b + \frac{\alpha_s}{\alpha_r} \times [(1 - \alpha_b) \times C_s + \alpha_b \times B(C_b, C_s)]$$

produces a result color that is a weighted average of the backdrop color, the source color, and the blended $B(C_b, C_s)$ term, with the weighting determined by the backdrop and source alphas α_b and α_s. For the simplest blend mode, **Normal**, defined by

$$B(c_b, c_s) = c_s$$

the compositing formula collapses to a simple weighted average of the backdrop and source colors, controlled by the backdrop and source alpha values. For more interesting blend functions, the backdrop and source alphas control whether the effect of the blend mode is fully realized or is toned down by mixing the result with the backdrop and source colors.

The result alpha, α_r, is actually a computed result, described below in Section 7.2.6, "Shape and Opacity Computations." The result color is normalized by the result alpha, ensuring that when this color and alpha are subsequently used to-