

Chapter 5

Attributes of Graphics Primitives

Myung-Soo Kim

Seoul National University

<http://cse.snu.ac.kr/mskim>

<http://3map.snu.ac.kr>

Attributes

- Color Attributes
- Point Attributes
- Line Attributes
- Curve Attributes
- Fill–Area Attributes
 - Fill Styles: Hollow, Solid, Patterned
- Character Attributes

RGB Color Components

TABLE 4-1

THE EIGHT RGB COLOR CODES FOR A THREE-BIT PER PIXEL FRAME BUFFER

<i>Color Code</i>	<i>Stored Color Values in Frame Buffer</i>			<i>Displayed Color</i>
	<i>RED</i>	<i>GREEN</i>	<i>BLUE</i>	
0	0	0	0	Black
1	0	0	1	Blue
2	0	1	0	Green
3	0	1	1	Cyan
4	1	0	0	Red
5	1	0	1	Magenta
6	1	1	0	Yellow
7	1	1	1	White

Color Lookup Tables

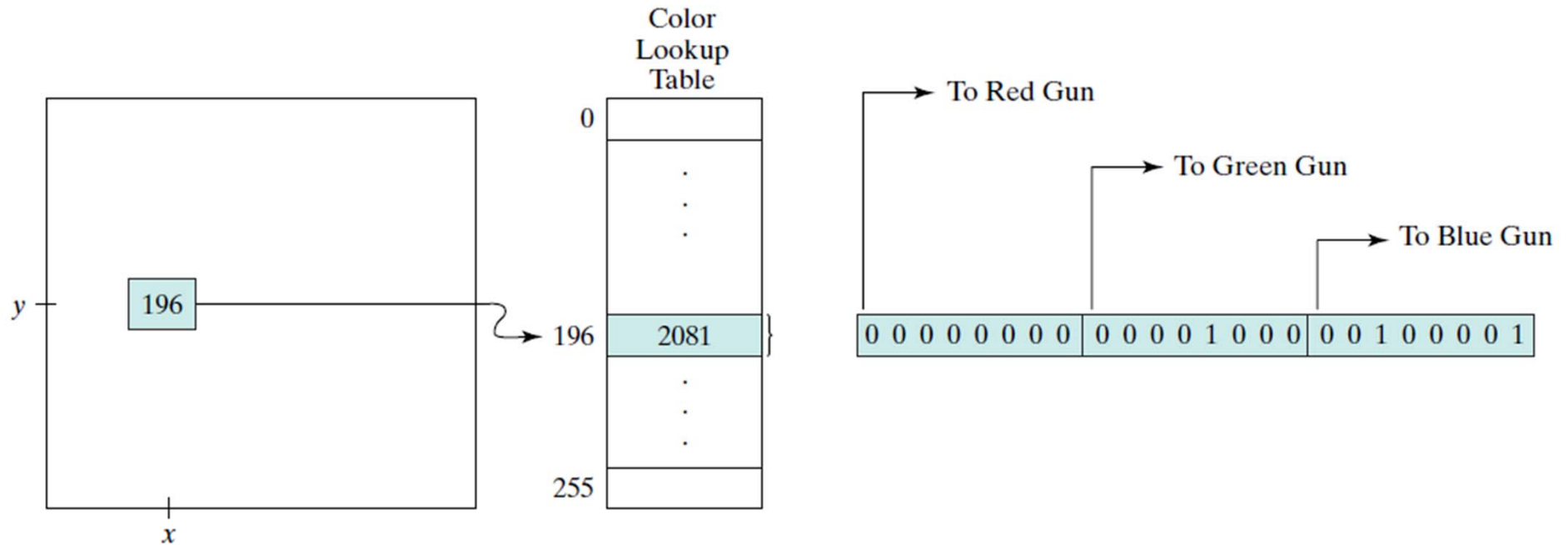


FIGURE 4-1 A color lookup table with 24 bits per entry that is accessed from a frame buffer with 8 bits per pixel. A value of 196 stored at pixel position (x, y) references the location in this table containing the hexadecimal value 0x0821 (a decimal value of 2081). Each 8-bit segment of this entry controls the intensity level of one of the three electron guns in an RGB monitor.

Color Blending

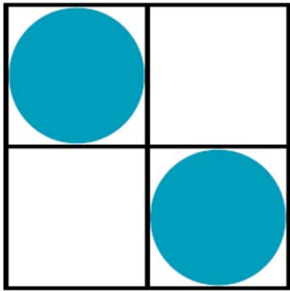
- The current color is the destination color
- The color of the second object is the source color

The new, blended color that is then loaded into the frame buffer is calculated as

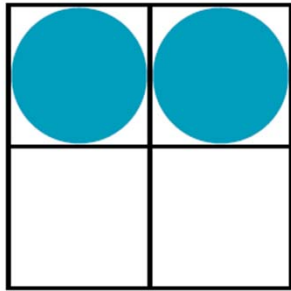
$$(S_r R_s + D_r R_d, S_g G_s + D_g G_d, S_b B_s + D_b B_d, S_a A_s + D_a A_d) \quad (4-1)$$

where the RGBA source color components are (R_s, G_s, B_s, A_s) , the destination color components are (R_d, G_d, B_d, A_d) , the source blending factors are (S_r, S_g, S_b, S_a) , and the destination blending factors are (D_r, D_g, D_b, D_a) . Computed values for the combined color components are clamped to the range from 0.0 to 1.0. That is, any sum greater than 1.0 is set to the value 1.0, and any sum less than 0.0 is set to 0.0.

Color-Blended Fill Regions

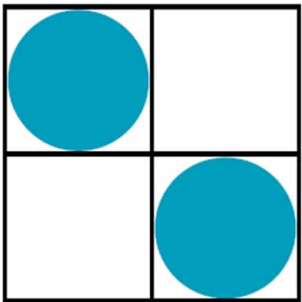
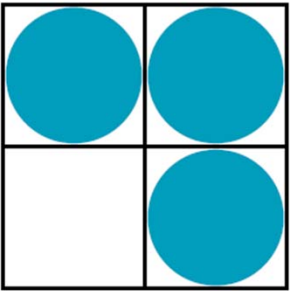
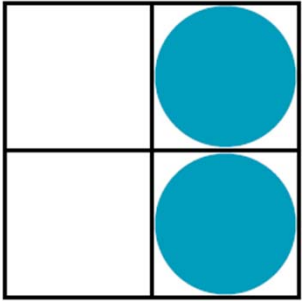
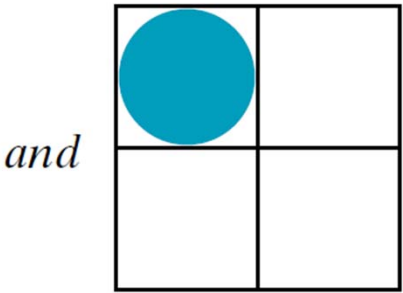


Pattern



Background

FIGURE 4-19 Combining a fill pattern with a background pattern using logical operations *and*, *or*, and *xor* (*exclusive or*), and using simple replacement.



Linear Soft-Fill Algorithm

The current RGB color \mathbf{P} of each pixel within the area to be refilled is some linear combination of \mathbf{F} and \mathbf{B} :

$$\mathbf{P} = t\mathbf{F} + (1 - t)\mathbf{B} \quad (4-2)$$

where the transparency factor t has a value between 0 and 1 for each pixel. For values of t less than 0.5, the background color contributes more to the interior color of the region than does the fill color. Vector Eq. 4-2 holds for each RGB component of the colors, with

$$\mathbf{P} = (P_R, P_G, P_B), \quad \mathbf{F} = (F_R, F_G, F_B), \quad \mathbf{B} = (B_R, B_G, B_B) \quad (4-3)$$

Similar color-blending procedures can be applied to an area whose foreground color is to be merged with multiple background color areas, such as a checkerboard pattern. When two background colors B_1 and B_2 are mixed with foreground color \mathbf{F} , the resulting pixel color \mathbf{P} is

$$\mathbf{P} = t_0\mathbf{F} + t_1\mathbf{B}_1 + (1 - t_0 - t_1)\mathbf{B}_2 \quad (4-5)$$