ATTRIBUTES OF OUTPUT PRIMITIVES
(Chapter 4 in *Computer Graphics*)

- Line Styles
- Color and Intensity
- Area Filling Commands
- Character Attributes
- Bundled Attributes
Line Styles

- line type
  - solid
  - dashed (short solid sections)
  - dotted (every other pixel)
  - set_linetype (lt)

- line width
  - dependent on the output device
    - parallel lines on a video monitor
    - pen change on a plotter
    - set_linewidth_scale_factor (lw)

- line color
  - set_line_color_index (lc)
    - lc indexes a color table
Color and Intensity

- range of choices
  - raster-scan systems:
    - large range
  - random-scan systems:
    - small range
## Color Tables

<table>
<thead>
<tr>
<th>COLOR CODE</th>
<th>STORED COLOR VALUES IN FRAME BUFFER</th>
<th>DISPLAYED COLOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RED</td>
<td>GREEN</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- 8 color choices with 3 bits per pixel
color lookup tables

- frame buffer contents are indices into the lookup table
- 6 bits per pixel means 64 choices
- 12 bits per color means 4096 colors
- set_color_table (ct, c)
  - ct is the color table position (0 to 63)
  - c is the color code
gray scale

- for monitors with no color capability

<table>
<thead>
<tr>
<th>INTENSITY CODES</th>
<th>STORED INTENSITY VALUES IN THE FRAME BUFFER (Binary Code)</th>
<th>DISPLAYED GRAY SCALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0 (00)</td>
<td>Black</td>
</tr>
<tr>
<td>0.33</td>
<td>1 (01)</td>
<td>Dark Gray</td>
</tr>
<tr>
<td>0.67</td>
<td>2 (10)</td>
<td>Light Gray</td>
</tr>
<tr>
<td>1.0</td>
<td>3 (11)</td>
<td>White</td>
</tr>
</tbody>
</table>
Area Filling

- scan-line algorithm
- antialiasing area boundaries
- boundary-fill algorithm
- flood-fill algorithm
- area-filling commands
scan-line algorithm

- intersect area boundaries and scan lines to identify pixels inside the area
scan-line algorithm, cont.

- for each nonhorizontal scan line
  - locate the intersection with each edge
  - order intersections (x,y) on y and then x
  - remove pairs of intersections
  - fill

- for vertices
  - record only one vertex for monotonic increasing or monotonic decreasing edges
  - record two vertices for local maxima or minima
take advantage of coherence

- adjacent pixels are usually alike
- adjacent scan lines are usually alike
maintain an active edge list

- sort polygon edges according to maximum y coordinates
- maintain pointers
  - Top points to the highest active edge
  - Bottom points to the lowest active edge
to produce a patterned fill

- modify scan-line procedures so that a selected pattern is superimposed
  - begin from a specified starting position

Pattern
Start Position
antialiasing area boundaries

- adjust intensity according to overlap
estimating overlap by subdivision

- subdivide the pixel and count subdivisions inside the boundary

- 2 subdivision centers inside the boundary suggests 50% intensity
estimating overlap using Pitteway-Watkinson

- modify the Bresenham algorithm and use the magnitude of the decision variable $p$

\[ y_i + 0.5 - \]
\[ y_i - \]
\[ y = m(x_i - 0.5) + b \]
\[ y_i - 0.5 - \]
\[ x_i - 0.5 \quad x_i \quad x_i + 0.5 \]

- overlap $= mx_i + b - y_i + 0.5$
estimating coverage by very small polygons

- use pixel subdivision
boundary-fill algorithm - an alternative to the scan-line method

- select a color and an interior point
  or
  select a pattern, a reference point and an interior point

- paint neighbors which are not painted and are not boundary points
  - all four neighbors
    or
  - all eight neighbors
making the boundary-fill algorithm more efficient

- fill the scan line containing the starting point
  - fill scan lines above
  - then fill scan lines below
- stack numbered pixels
flood-fill algorithm

- specify an interior color value that is to be replaced by the fill color
Area-filling Commands

- fill style
  - solid, hollow, patterned, etc.
  - set_fill_area_interior_style (fs)

- fill color
  - set_fill_area_color_index (fc)

- fill pattern
  - set_fill_pattern_index (pi)

- create pattern
  - set_pattern_representation (pi, nx, ny, cp)
    - pi = pattern index number
    - nx = x dimension of the pattern
    - ny = y dimension of the pattern
    - cp = two-dimensional nx-by-ny array
    - example: cp[3,6] := 6

- position pattern
  - set_pattern_reference_point (xp, yp)

(a) fill pattern
(b) patterned display
Character Attributes

- text attributes
  - text font
    - set_text_font (tf)
  - text color
    - set_text_color_index (tc)
  - text size (maintaining aspect ratio)
    - set_character_height (ch)
  - character string orientation
    - set_character_up_vector (dx, dy)
text attributes, continued

- character string direction
  - set_text_path (tp)
- tp is assigned
  right
  down
  left
  or
  up
text attributes, continued

- combining up_vector and text_path specifications

Direction of Character up Vector (a)

Text Path Direction (b)

RIGHT STRING (a)

SLICE (b)
text attributes, continued

- alignment
  - set_text_alignment (h, v)
    - h is assigned left, right, or center
    - v is assigned top or bottom

\[
\begin{array}{|c|c|}
\hline
T & A \hspace{1cm} \text{RIGHT} \\
O & L \hspace{1cm} \text{ALIGNMENT} \\
P & I \hspace{1cm} \text{ALIGNMENT} \\
G & B \hspace{1cm} \text{ALIGNMENT} \\
N & O \hspace{1cm} \text{ALIGNMENT} \\
M & T \hspace{1cm} \text{ALIGNMENT} \\
\hline
\end{array}
\]
marker attributes are similar to text attributes

- set_marker_type (mt)
- set_marker_color_index (mc)
- set_marker_size_scale_factor (ms)
inquiry functions

- check current settings of attribute values

- examples
  - inquire_linetype (ilt)
  - inquire_fill_area_color_index (ifc)

- use: inquire_linetype (old_style);
  set_linetype (new_style);
  
  set_linetype (old_style);
Bundled Attributes

- unbundled attributes
  - each attribute is defined explicitly
  - possibly device dependent

- bundled attributes
  - provides interpretation of attribute settings for different devices using a bundle table
  - the program must specify the output device
  - activate_workstation (ws)

<table>
<thead>
<tr>
<th>WORKSTATION TYPE CODE</th>
<th>WORKSTATION DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raster Color Monitor with Keyboard</td>
</tr>
<tr>
<td>2</td>
<td>Raster Color Monitor with Keyboard and Graphics Tablet</td>
</tr>
<tr>
<td>3</td>
<td>DVST with Keyboard and Thumb Wheels</td>
</tr>
<tr>
<td>4</td>
<td>Vector Refresh Monitor with Keyboard</td>
</tr>
<tr>
<td>5</td>
<td>Color Plotter</td>
</tr>
</tbody>
</table>
line attributes

- create a bundle table for each workstation
  - set_line_representation (ws, li, lt, lw, lc)

- reference it with the line index parameter li
  - set_line_index (li)
color and intensity attributes

- create a bundle table for each workstation
  - set_color_representation (ws, ci, r, g, b)

- reference it with the color index parameter ci
  - set_color_index (ci)

<table>
<thead>
<tr>
<th>COLOR INDEX CI</th>
<th>RED COMPONENT (R)</th>
<th>GREEN COMPONENT (G)</th>
<th>BLUE COMPONENT (B)</th>
<th>COLOR DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Black</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>Shades of Red</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.25</td>
<td>0</td>
<td>Shades of Green</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.50</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.75</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
</tbody>
</table>
area-filling attributes

- create a bundle table for each workstation
  - set_fill_area_representation (ws, fi, fs, fc, pi)

- reference it with the fill index parameter fi
  - set_fill_area_index (fi)
pattern attributes

- set_pattern_representation (ws, fpi, nx, ny, cp);
  - cp is the nx-by-ny array
  - fpi is the fill pattern index
- set_pattern_reference_point (xp, yp);
- referenced via the area-filling bundle table
text attributes

- create a bundle table for each workstation
  - set_text_representation (ws, ti, tf, te, ts, tc)
- reference it with the text index parameter ti
  - set_text_index (ti)
marker attributes

- create a bundle table for each workstation
  - set_marker_representation (ws, mi, mt, ms, mc)

- reference it with the marker index parameter mi
  - set_marker_index (mi)
ATTRIBUTES OF OUTPUT PRIMITIVES

- Line Styles
- Color And Intensity
- Area Filling Commands
- Character Attributes
- Bundled Attributes