

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

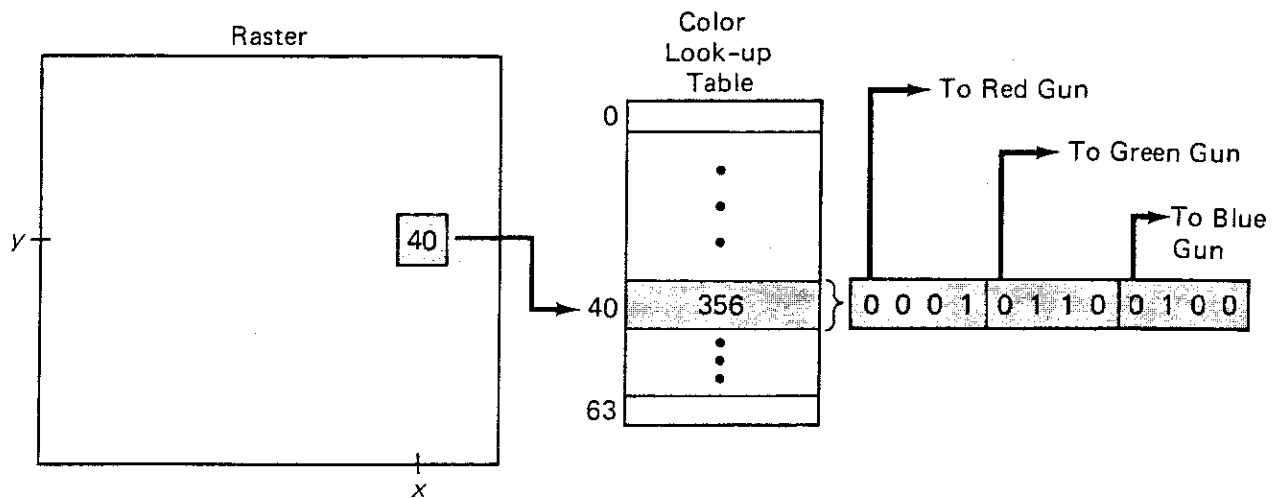
- range of choices
 - raster-scan systems:
 - large range
 - random-scan systems:
 - small range

color tables

COLOR CODE	STORED COLOR VALUES IN FRAME BUFFER			DISPLAYED COLOR
	RED	GREEN	BLUE	
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

- 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

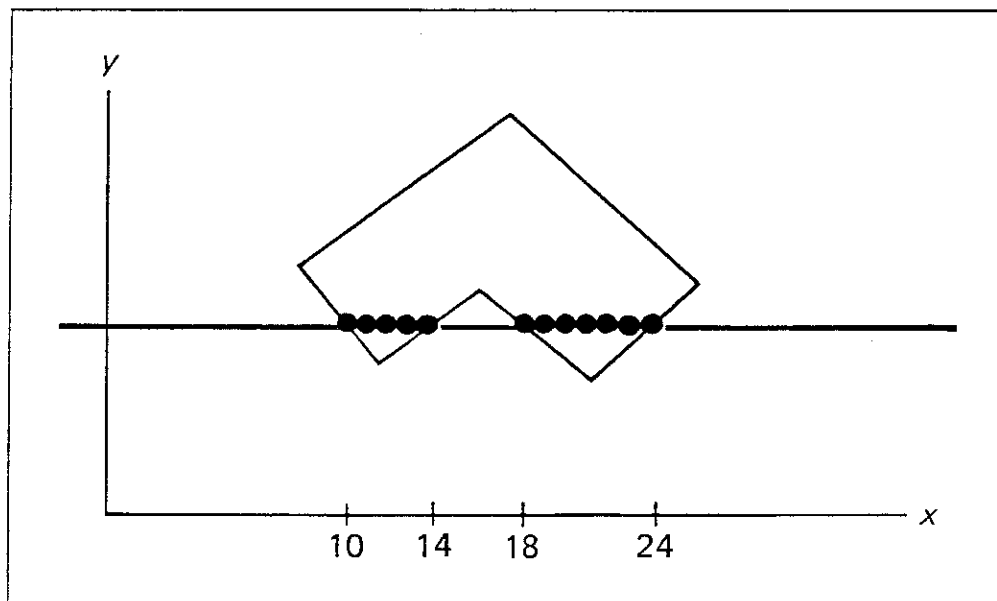
INTENSITY CODES	STORED INTENSITY VALUES IN THE FRAME BUFFER (Binary Code)		DISPLAYED GRAY SCALE
0.0	0	(00)	Black
0.33	1	(01)	Dark Gray
0.67	2	(10)	Light Gray
1.0	3	(11)	White

Area Filling

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

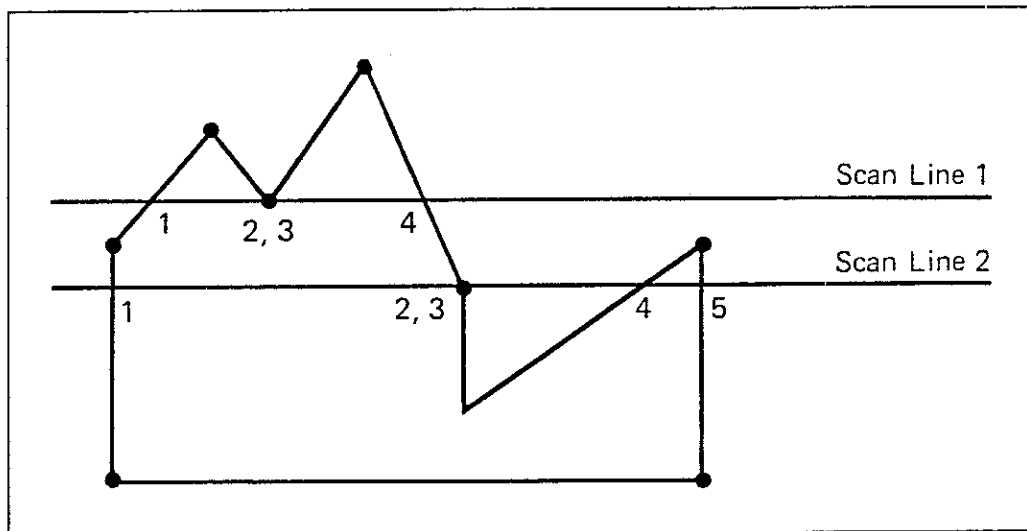
scan-line algorithm

- intersect area boundaries and scan lines to identify pixels inside the area



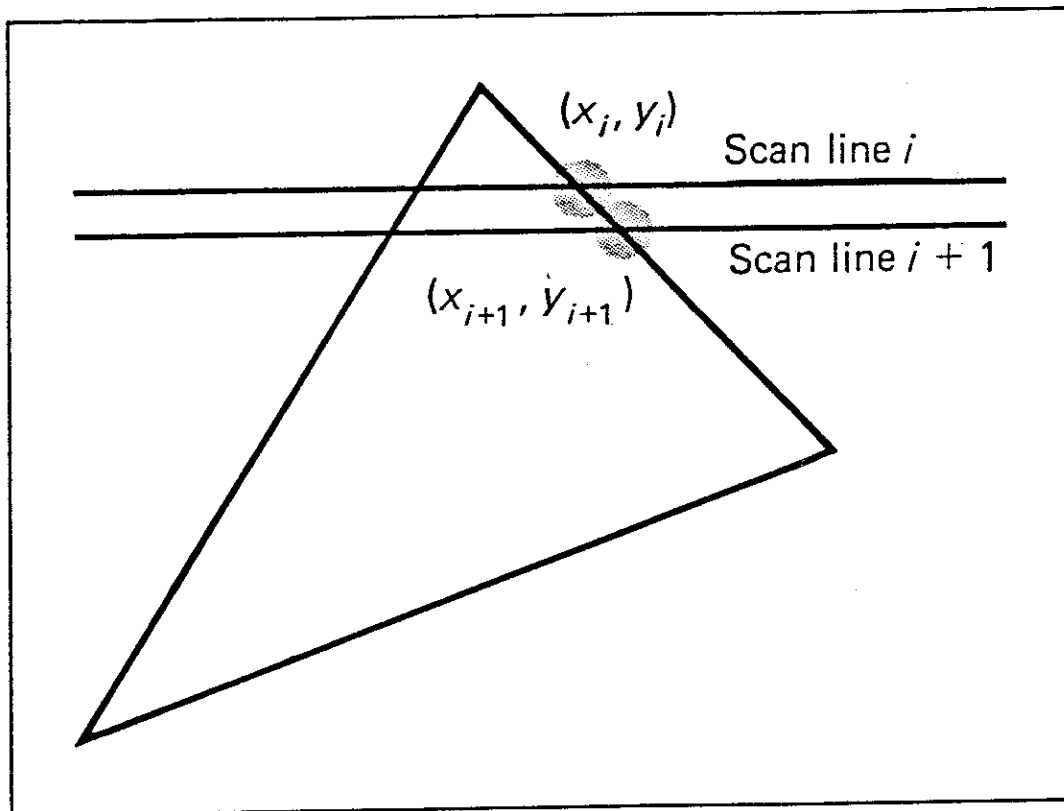
scan-line algorithm, cont.

- 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



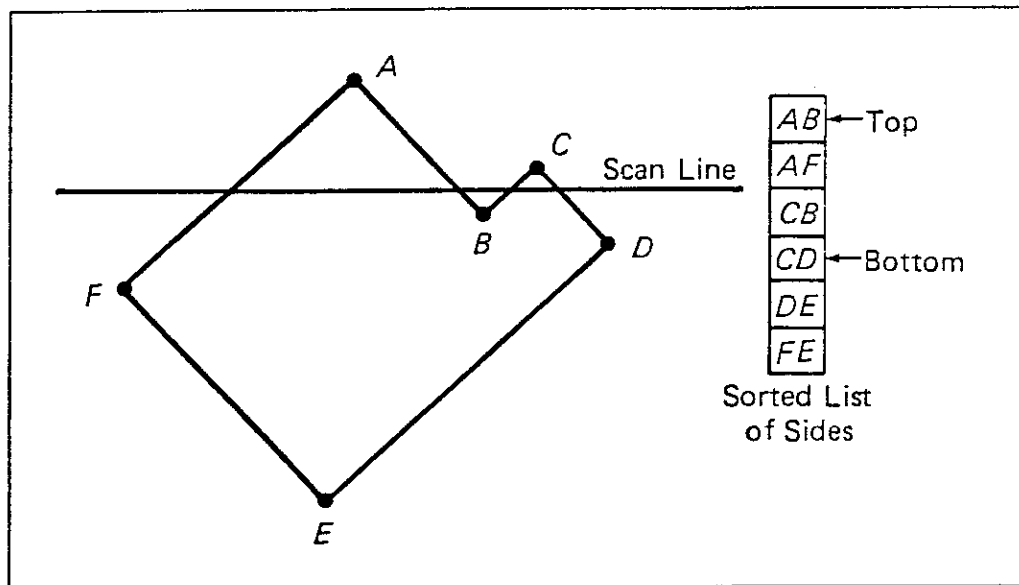
take advantage of coherence

- adjacent pixels are usually alike
- adjacent scan lines are usually alike



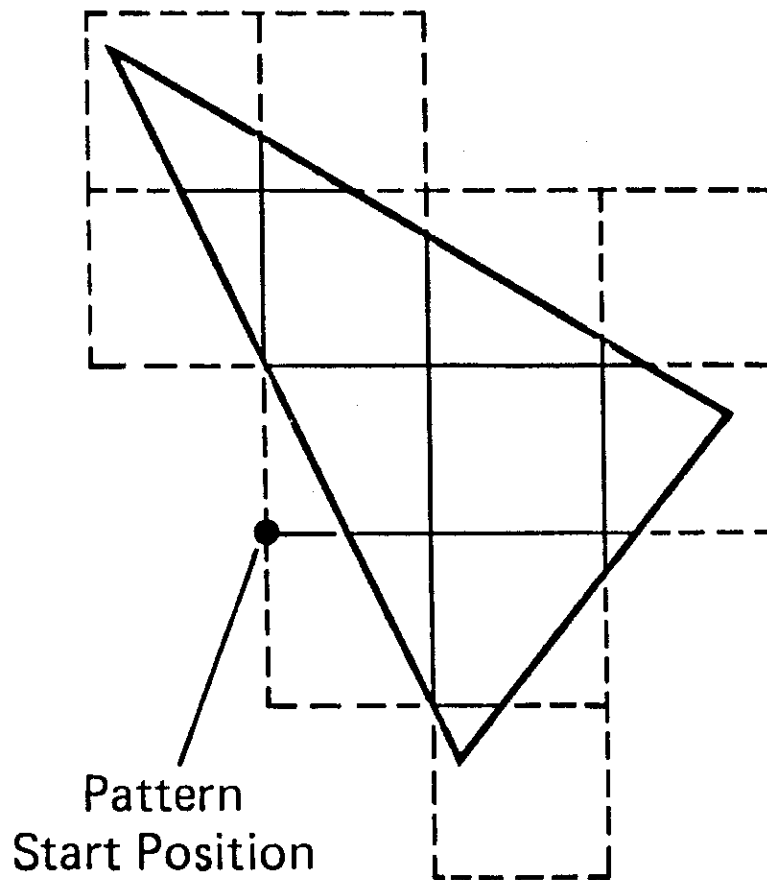
maintain an active edge list

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



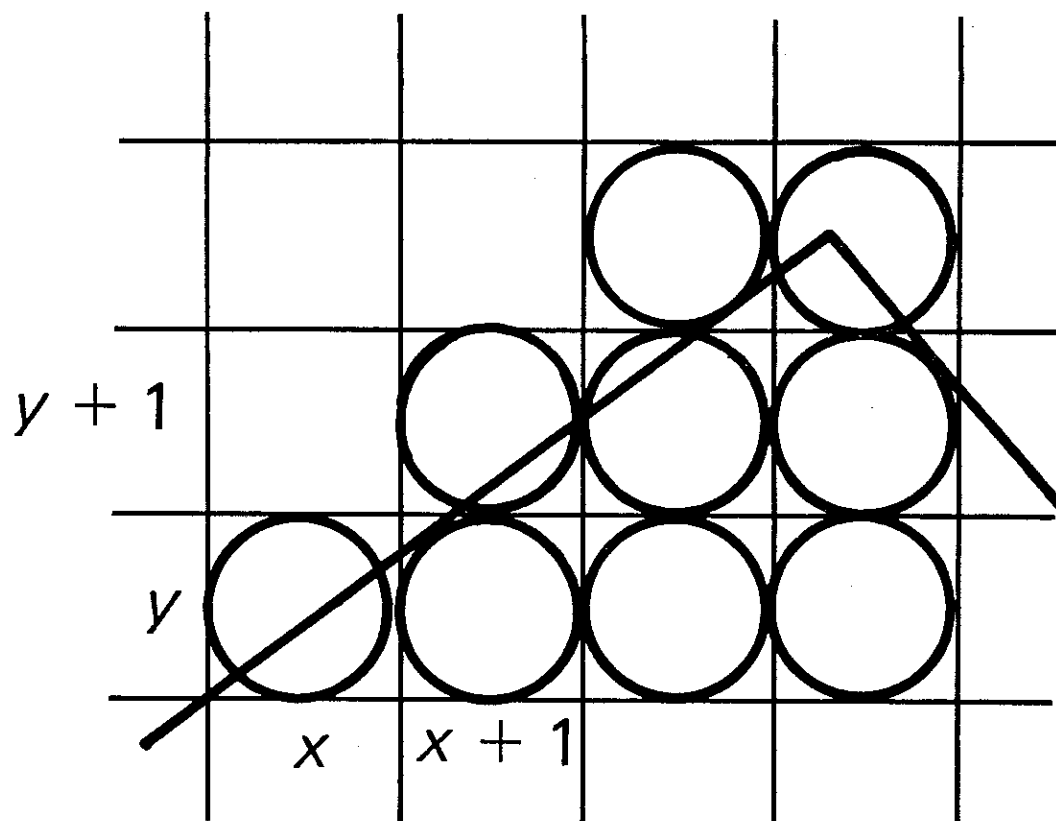
to produce a patterned fill

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



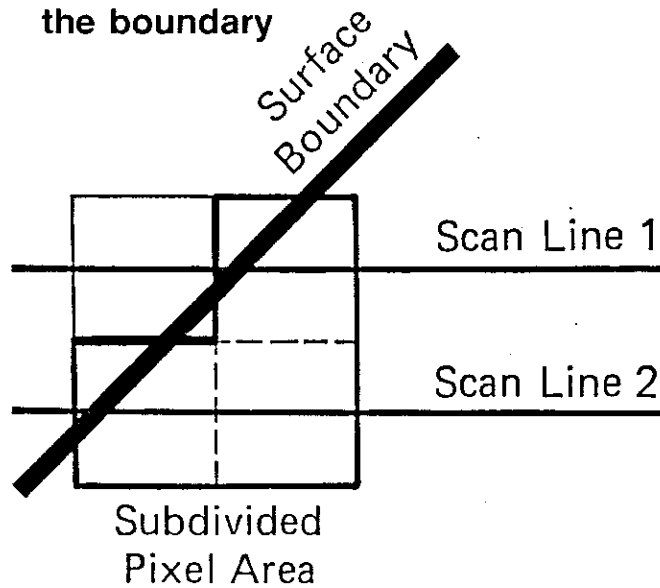
antialiasing area boundaries

- adjust intensity according to overlap



estimating overlap by subdivision

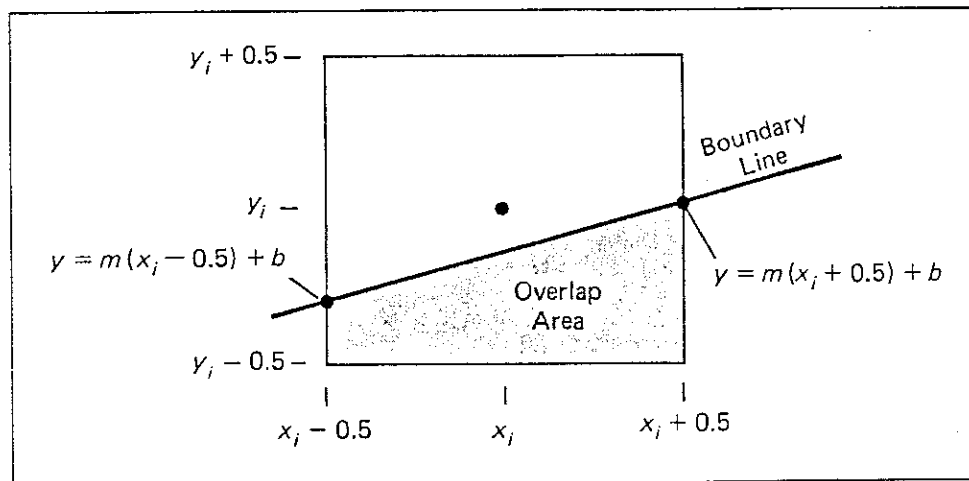
- subdivide the pixel and count subdivisions inside the boundary



- 2 subdivision centers inside the boundary suggests 50% intensity

estimating overlap using Pitteway-Watkinson

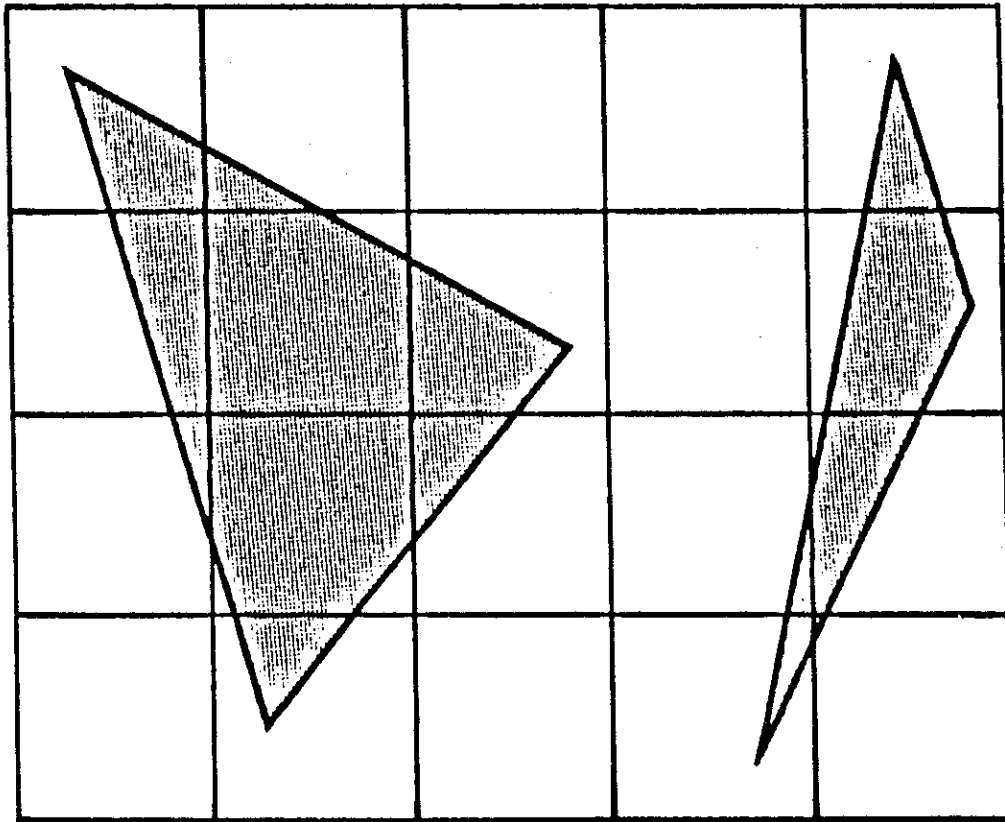
- modify the Bresenham algorithm and use the magnitude of the decision variable p



- $\text{overlap} = mx_i + b - y_i + 0.5$

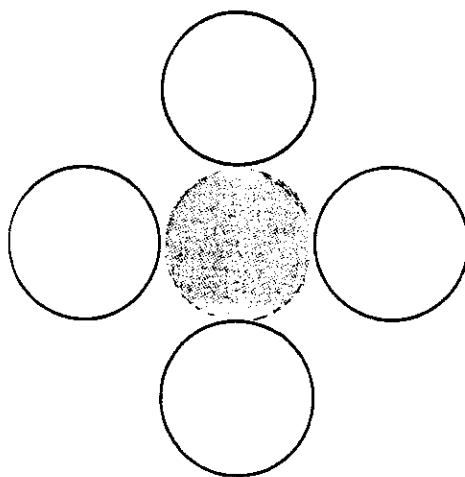
estimating coverage by very small polygons

- use pixel subdivision

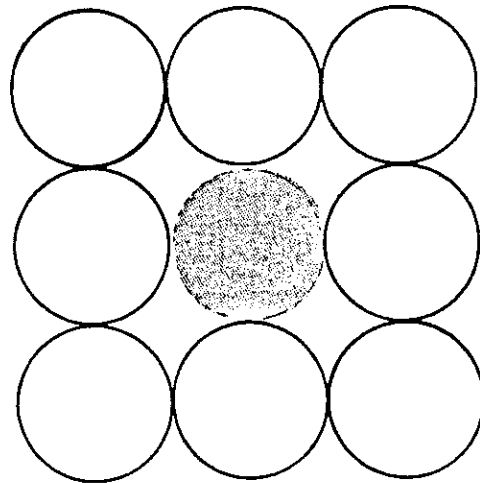


boundary-fill algorithm - an alternative to the scan-line method

- 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



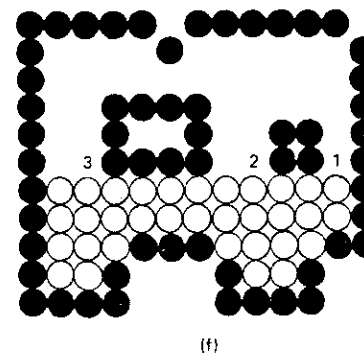
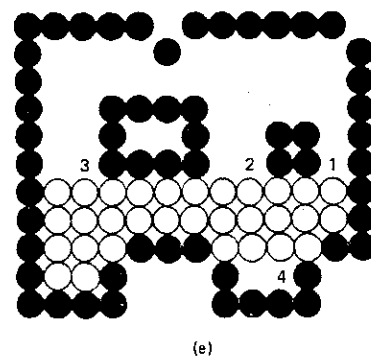
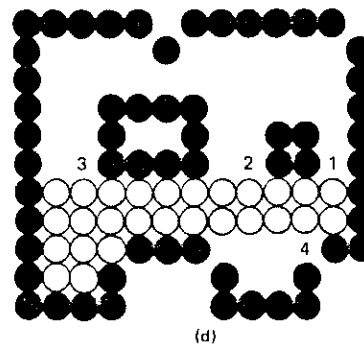
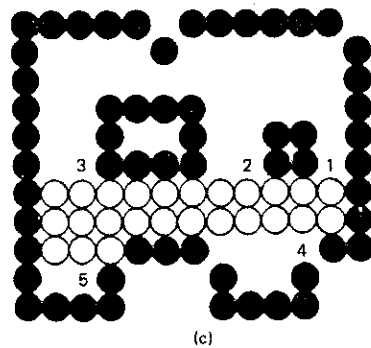
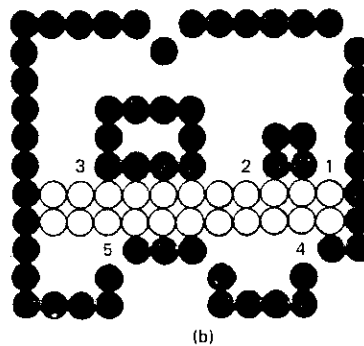
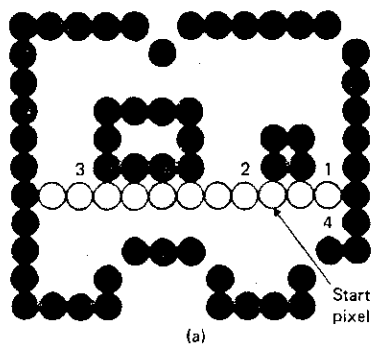
(a)



(b)

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

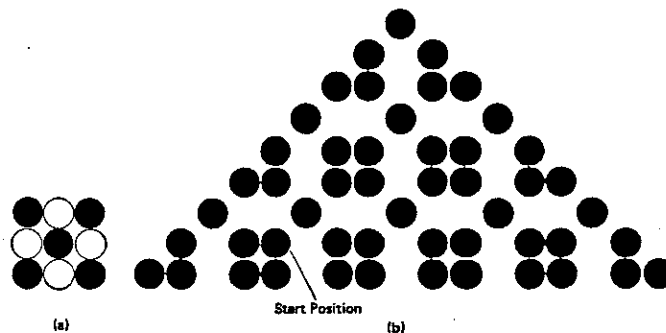
- specify an interior color value that is to be replaced by the fill color

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)

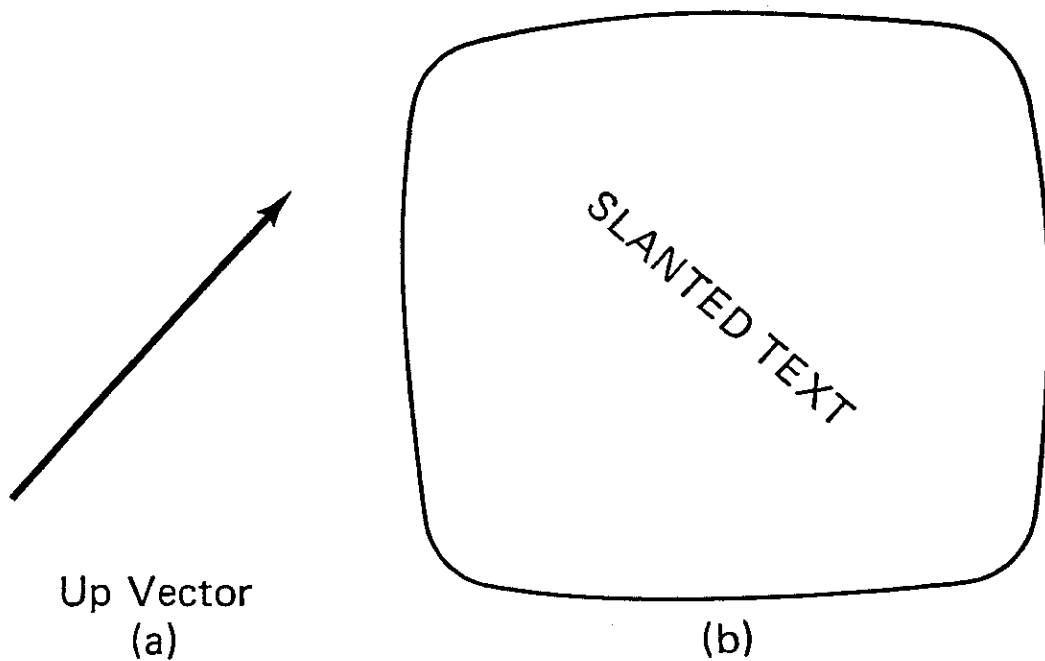
(a) fill pattern

(b) patterned display



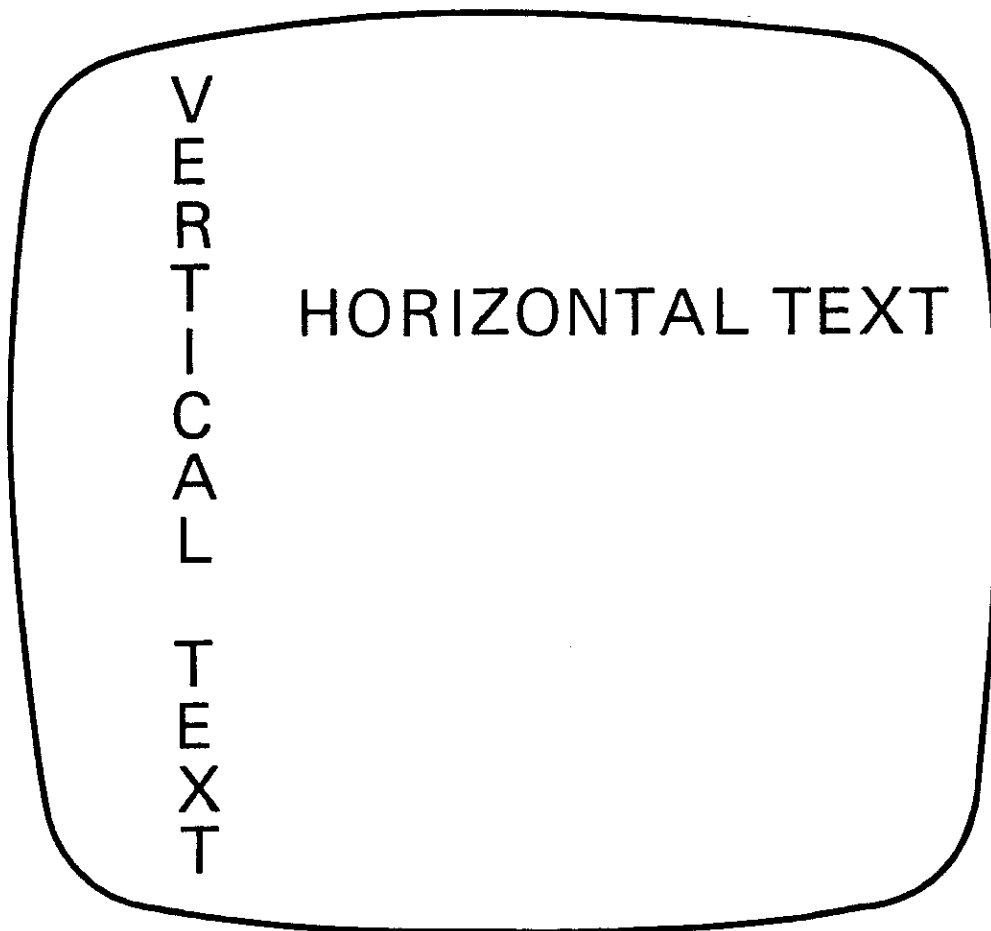
Character Attributes

- 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)`



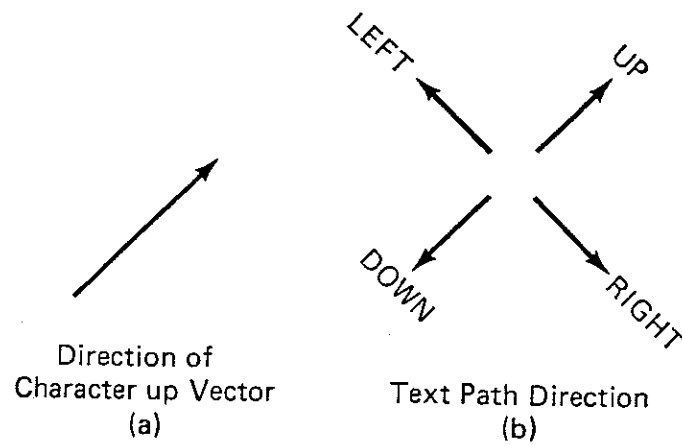
text attributes, continued

- character string direction
 - set_text_path (tp)
 - tp is assigned
 - right
 - down
 - left
 - or
 - up



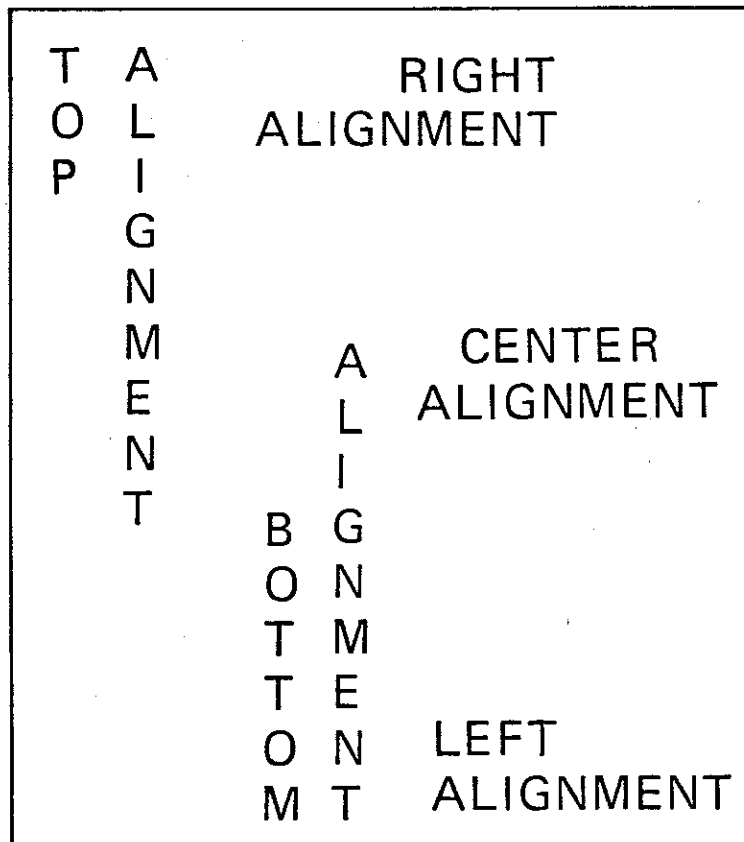
text attributes, continued

- combining up_vector and text_path specifications



text attributes, continued

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



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)`

WORKSTATION TYPE CODE	WORKSTATION DESCRIPTION
1	Raster Color Monitor with Keyboard
2	Raster Color Monitor with Keyboard and Graphics Tablet
3	DVST with Keyboard and Thumb Wheels
4	Vector Refresh Monitor with Keyboard
5	Color Plotter

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)

COLOR INDEX CI	RED COMPONENT (R)	GREEN COMPONENT (G)	BLUE COMPONENT (B)	COLOR DESCRIPTION
0	0	0	0	Black
1	0.25	0	0	Shades of Red
2	0.50	0	0	
3	0.75	0	0	
4	1.0	0	0	
5	0	0.25	0	Shades of Green
6	0	0.50	0	
7	0	0.75	0	
8	0	1.0	0	
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

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**