# Antialiasing

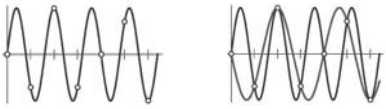
- Aliasing
- Supersampling
- Antialiased Bresenham
- Area sampling (Unweighted & Weighted)
- Gupta-Sproull



## Aliasing

Aliasing in Signal Processing:

Unwanted artifacts caused by undersampling signals



sampled signal reconstructed signal



## Aliasing

Aliasing in Computer Graphics:

- Unwanted artifacts in representation of continuous primitives with discrete samples (pixels)
- Caused by finite addressability of the display
- Evidenced as Moiré Patterns, Jaggies or Staircasing



## Increasing Resolution

In signal processing:

- **Nyquist's theorem:** The sampling rate must be equal to, or greater than, twice the highest frequency component in the analog signal

In Computer graphics:

- Higher addressability (Screen resolution): currently is ~72dpi
- To eliminate aliasing: must exceed the resolution of human eye (~600dpi)
- Expensive: Doubling resolution in x and y costs 4 times memory, memory bandwidth and scan conversion time

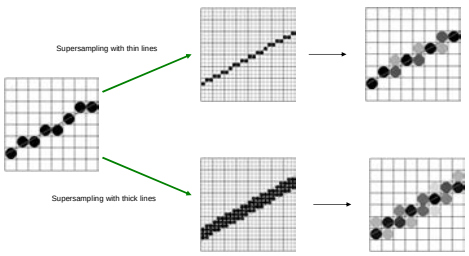


## Supersampling

- Construct a virtual higher resolution display by expanding each pixel into k×k subpixels
- Scan convert the desired graphic into the above
- Set the intensity of each actual display pixel to a value that depends upon the number of active subpixels



## Supersampling (2)



## Supersampling (3)

- Nonuniform pixel weighting masks can be employed to give the subpixels at the center more weight than those near the boundary.


- Foreground and background color?



## Supersampling (4) - Color blending

Linear blend between the foreground and background color

$\alpha = {}^{n}/_{N}$, $0 < \alpha < 1$

color = $\alpha F + (1- \alpha)B$

RGB:

$R = \alpha R_F + (1- \alpha)R_B$

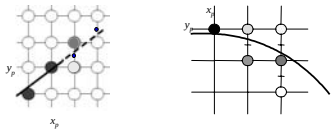
$G = \alpha G_F + (1- \alpha)G_B$

$B = \alpha B_F + (1- \alpha)B_B$



## Antialiased Bresenham

- An antialiased line has a series of virtual pixels each located at the proper address.
- Use the distance values to determine pixel intensities.

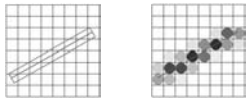



- Problem: Non-uniform width along the line/curve
  - Refined version: Gupta-Sproull Antialiased lines



## Area Sampling

- Consider lines to have width and area
- Set each pixel's intensity proportional to the area of the pixel covered by the line



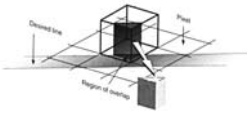
- Remember how to compute the intersection area?
- A rough approximation formulated by dividing each pixel into a finer grid of pixels



## Unweighted Area Sampling

Properties

- Pixel intensity decreases as distance from the line center increases.
- Pixels that do not overlap with the line area are not changed.
- Equal areas contribute equal intensity, regardless of their relative positions to the line.

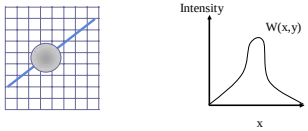



Box filter for square pixels



## Weighted Area Sampling

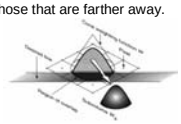
- Intuitively, pixel cut through the center should be more heavily weighted than one cut along corner with the same area
- Consider pixels as circular areas larger than the square tile (typically with diameter of two pixel units)
- Greater weight if closer to center of pixel



## Weighted Area Sampling (2)

- Volume is a circular cone
  - Normalize so that volume = 1
- Properties
  - Pixel intensity decreases as distance from the line center increases.
  - Pixels that do not overlap with the line area are not changed.
  - Equal areas can have unequal contributions: those closer to the line center are weighted more than those that are farther away.
  - Rotational symmetry

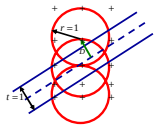
Cone filter for circular pixel with diameter of two grid units





## Gupta-Sproull Antialiased Lines

- Standard Bresenham to choose E or NE
- Incrementally compute distance D from chosen pixel to center of line
- Vary pixel intensity by value of D
- Do this for line above and below






## Gupta-Sproull Antialiased Lines (2)

- Use coarse (4-bit, say) lookup table for intensity
- *Filter* value depends *only* on D (distance) and *t (a constant)*, not the slope of line**!** (Very cleaver)
- For *line_width*, $t = 1$ geometry and associated calculations greatly simplify