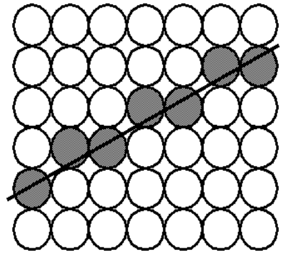
Overview

- Antialiasing Techniques
  - Super sampling
  - Area sampling
    - unweighted
    - weighted
- Clipping
  - Cohen-Sutherland line clipping algorithm
  - Liang-Barsky line clipping algorithm
  - Sutherland-Hogeman polygon clipping

Antialiasing

Aliasing, jagged edges or staircasing can be reduced by:

- Higher screen resolution
  - Need a huge frame buffer
- Antialiasing techniques
  - Vary pixel intensities along to smooth the edge.



Antialiasing Techniques

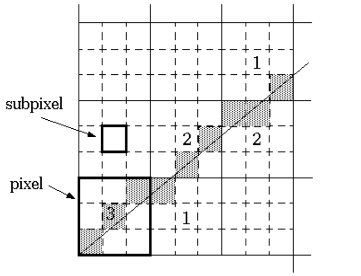
- Super Sampling
  - Compute intensities at sub-pixel grid positions and combine the results to obtain the pixel intensity.
- Unweighted Area Sampling
  - Find pixel intensity by calculating the areas of overlap of each pixel within the objects to be displayed.
  - Pixel intensity is proportional to the amount of area covered.

Antialiasing Techniques

- Weighted Area Sampling
  - Define a weighting function that determines the influence on the intensity of the pixel.
- Pixel Phasing
  - Lines are smoothed by moving the electron beam to a closer approximate of the mathematical line.

Supersampling (zero line width)

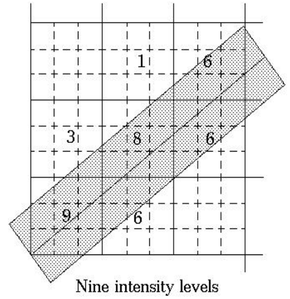
- Example: a straight line on a gray scale display
- Divide each pixel into sub-pixels.
- The number of intensities are the max number of sub-pixels selected on the line segment within a pixel.



Subpixels selected by Bresenham's algorithm

Supersamling (finite line width)

- The intensity level for each pixel is proportional to the number of sub-pixels inside the polygon representing the line area.
- Line intensity is distributed over more pixels.



Nine intensity levels

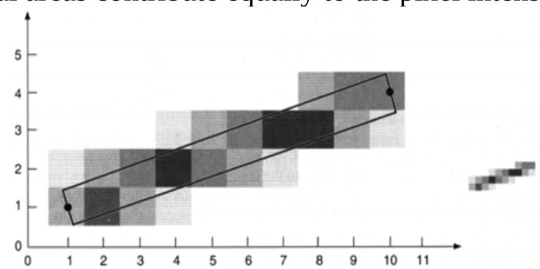
Supersamling (finite line width)

Disadvantages

- More calculations involved to identify interior pixels.
- Positioning of the line depends on the slope of the line.
  - 45° - line centered in polygon
  - Horizontal or vertical line
    - line path on polygon boundary
  - $|m| < 1$
    - line path closer to lower boundary
  - $|m| > 1$
    - line path closer to upper boundary

Unweigted Area Sampling

1. The intensity of a pixel decreases as the distance between the pixel center and the edge increases.
2. The primitive must intersect the pixel to have some effect.
3. Equal areas contribute equally to the pixel intensity.



Intensity of a pixel is proportional to its area covered by the line

Weighted Area Sampling

- Equal areas can contribute to unequal intensity. (We change property 3).
- Circular pixel geometry.

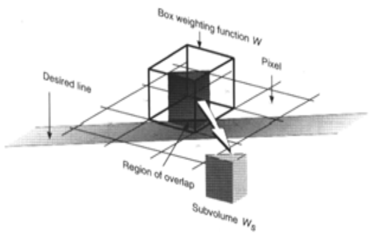
Weighting (Filter) Function

- Determines the influence on the intensity of a pixel of a given small area dA of a primitive.
- This function is constant for unweighted and decreases with increasing distance for weighted.
- Total intensity is the integral of the weighting (filter) function over the area of overlap.
- W_s is the volume (always between 0 and 1)
- $I = I_{max} \bullet W_s$

Unweighted Area Sampling

<u>Box Filter</u>:

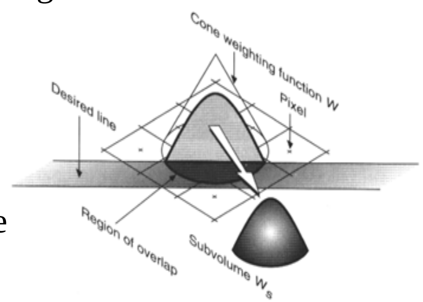
- W_s is a wedge of the box.
- Height of the box normalizes to 1 (box volume = 1)
- A thick line covering the entire pixel has intensity:
- $I = I_{max} \bullet 1 = I_{max}$



Weighted Area Sampling

<u>Cone Filter</u>:

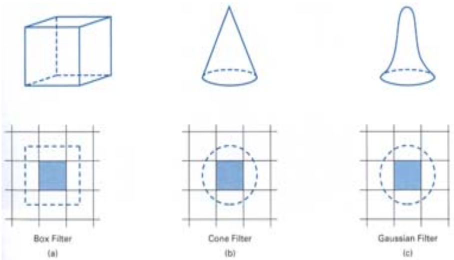
- A circular cone, where the base is the radius of the unit distance of the integer grid.
- Rotational symmetry.
- Linear decrease of the function with radial distance.
- Normalized to 1 (volume under entire cone is 1)



Filter Functions

- Optimal filters are computationally more expensive.
- Cone filters are a very reasonable compromise between cost

  and quality.



Anti-Aliasing

Pixel Phasing:

- Pixel positions can be shifted by a fraction of a pixel diameter (1/4, 1/2, or 3/4) to plot points closer to the mathematical line.

Line Intensity Differences:

- The diagonal line appears less bright than the horizontal. (The diagonal line is longer than the horizontal line by a factor of sqrt(2)).
  - Total intensity is proportional to their length.

Clipping Algorithms

Line Clipping:

- Cohen-Suterland (encoding)
  - Oldest and most commonly used
- Nicholl-Lee-Nicholl (encoding) (more efficient)
- Cyrus-Beck and Liang-Barsky (parametric)
  - More efficient than Cohen-Sutherland

Polygon Clipping:

- Sutherland-Hodgeman (divide and conquer strategy)
- Weiler-Atherton (modified for concave polygons)

Cohen-Sutherland Line-Clipping

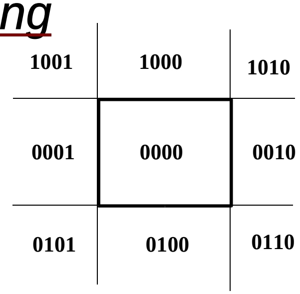
1. Encode end points

Bit 0 = point is left of window
Bit 1 = point is right of window
Bit 2 = point is below window
Bit 3 = point is above window

2. If $C_1 \wedge C_2 \neq 0$ then P_1P_2 is trivially rejected

3. If $C_1 \vee C_2 = 0$ then P_1P_2 is trivially accepted

4. Otherwise subdivide and go to step 1 with new segment.



C_1 = Bit code of P1

C_2 = Bit code of P2

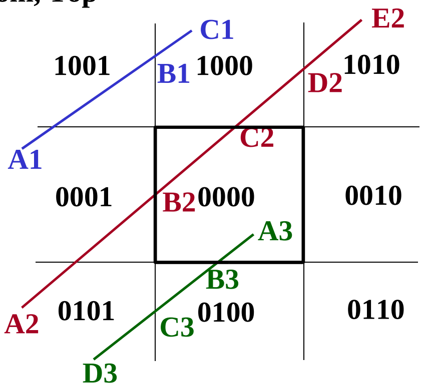
Cohen-Sutherland Line-Clipping

Clip order: Left, Right, Bottom, Top

1) A1C1
2) B1C1
3) reject

1) A2E2
2) B2E2
3) B2D2
4) B2C2
5) accept

1) A3D3
2) A3C3
3) A3B3
4) accept



Cohen-Sutherland Line-Clipping

- Will do unnecessary clipping.
- Not the most efficient.
- Clipping and testing are done in fixed order.
- Efficient when most of the lines to be clipped are either rejected or accepted (not so many subdivisions).
- Easy to program.
- Parametric clipping are the most efficient. (Liang-Barsky and Cyrus-Beck)

Liang-Barsky Line-Clipping

- More efficient than Cohen-Sutherland
- Clipping conditions:
  - A line is inside the clipping region for values of t such that:

$$x_{\min} \le x_1 + t\Delta x \le x_{\max} \qquad \Delta x = x_2 - x_1$$

$$y_{\min} \le y_1 + t\Delta y \le y_{\max} \qquad \Delta y = y_2 - y_1$$

Liang-Barsky Line-Clipping

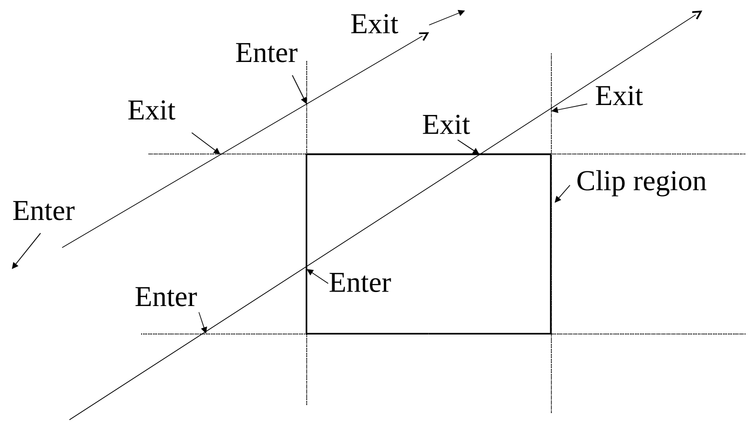
- The infinitely line intersects the clip region edges when:

$$t_k = \frac{q_k}{p_k} \quad \text{where} \quad
\begin{array}{lll}
p_1 = -\Delta x & q_1 = x_1 - x_{\min} & \text{Left boundary} \\
p_2 = \Delta x & q_2 = x_{\max} - x_1 & \text{Right boundary} \\
p_3 = -\Delta y & q_3 = y_1 - y_{\min} & \text{Bottom boundary} \\
p_4 = \Delta y & q_4 = y_{\max} - y_1 & \text{Top boundary}
\end{array}$$

Liang-Barsky Line-Clipping

- When $p_k < 0$, as t increases line goes from outside to inside - entering
- When $p_k > 0$, line goes from inside to outside - exiting
- When $p_k = 0$, line is parallel to an edge
- If there is a segment of the line inside the clip region, a sequence of infinite line intersections must go: entering, entering, exiting, exiting

Liang-Barsky Line-Clipping

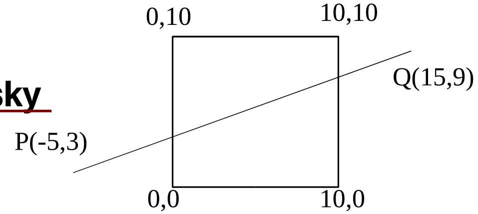


Liang-Barsky Line-Clipping

- Set $t_{min} = 0$ and $t_{max} = 1$.
- Calculate the $\boldsymbol{t}$ values:
  - If $t < t_{min}$ or $t > t_{max}$ ignore it.
  - Otherwise classify the $\boldsymbol{t}$ values as entering or exiting.
- If $t_{min} < t_{max}$ then draw a line from:

$$(x_1 + \Delta x \cdot t_{min}, y_1 + \Delta y \cdot t_{min}) \text{ to } (x_1 + \Delta x \cdot t_{max}, y_1 + \Delta y \cdot t_{max})$$

Example Liang-Barsky



$$t_{left} = \frac{q_1}{p_1} = \frac{x_1 - x_{min}}{-\Delta x} = \frac{-5-0}{-(15-(-5))} = \frac{1}{4}$$

Entering → $t_{min} = 1/4$

$$t_{right} = \frac{q_2}{p_2} = \frac{x_{max} - x_1}{\Delta x} = \frac{10-(-5)}{15-(-5)} = \frac{3}{4}$$

Exiting → $t_{max} = 3/4$

$$t_{bottom} = \frac{q_3}{p_3} = \frac{y_1 - y_{min}}{-\Delta y} = \frac{3-0}{-(9-3)} = -\frac{1}{2}$$

$t < 0$ then ignore

$$t_{top} = \frac{q_4}{p_4} = \frac{y_{max} - y_1}{\Delta y} = \frac{10-3}{9-3} = \frac{7}{6}$$

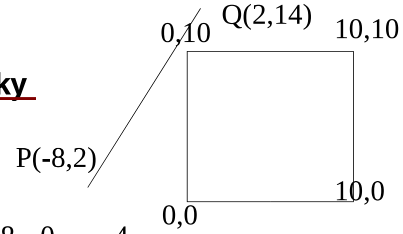
$t > 1$ then ignore

Liang-Barsky Line-Clipping

- We have t_{min} = 1/4 and t_{max} = 3/4
- Q-P = (15+5,9-3) = (20,6)

  $\downarrow$ Δx $\downarrow$ Δy
- If $t_{min} < t_{max}$, there is a line segment
  - compute endpoints by substituting t values
- Draw a line from (-5+(20)·(1/4), 3+(6)·(1/4)) to (-5+(20)·(3/4), 3+(6)·(3/4))

Example Liang-Barsky



$$t_{left} = \frac{q_1}{p_1} = \frac{x_1 - x_{\min}}{-\Delta x} = \frac{-8-0}{-(2-(-8))} = \frac{4}{5}$$

Entering → t_{min} = 4/5

$$t_{right} = \frac{q_2}{p_2} = \frac{x_{\max} - x_1}{\Delta x} = \frac{10-(-8)}{2-(-8)} = \frac{9}{5}$$

$t > 1$ then ignore

$$t_{bottom} = \frac{q_3}{p_3} = \frac{y_1 - y_{\min}}{-\Delta y} = \frac{2-0}{-(14-2)} = -\frac{1}{6}$$

$t < 0$ then ignore

$$t_{top} = \frac{q_4}{p_4} = \frac{y_{\max} - y_1}{\Delta y} = \frac{10-2}{14-2} = \frac{2}{3}$$

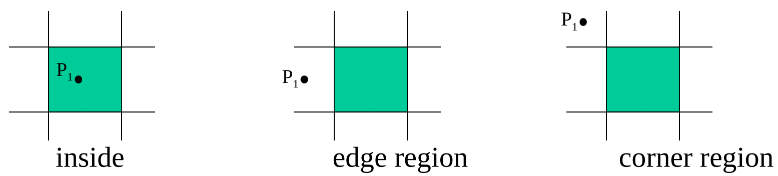
Exiting → t_{max} = 2/3

Liang-Barsky Line-Clipping

- We have t_{min} = 4/5 and t_{max} = 2/3
- Q-P = (2+8,14-2) = (10,12)
- $t_{min} > t_{max}$, there is no line segment do draw

Nicholl-Lee-Nicholl Line Clipping

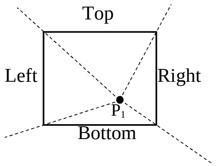
- Avoids multiple clipping of an individual line by creating more regions.
- Only three regions need to be considered.



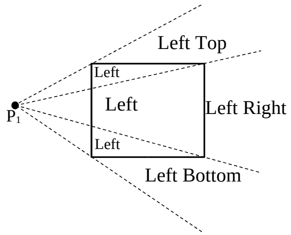
- Find position of P_2 relative to P_1.

Nicholl-Lee-Nicholl Line Clipping

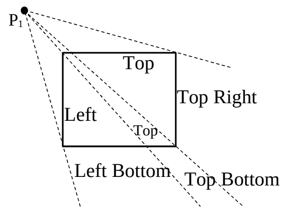
If P_1 inside and P_2 outside:



If P_1 is to the left:



If P_1 is to the left and above:



Nicholl-Lee-Nicholl Line Clipping

- To find which region P_2 is in, compare the slope of the line to the slopes of the clip rectangle.
- If P_1 is left of clip rectangle, then P_2 is in region Left Top if:

$$\text{slope}P_1P_{TR} < \text{slope}P_1P_2 < \text{slope } P_1P_{TL}$$

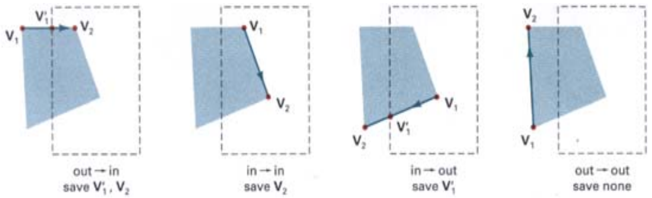
- Number of cases explodes in 3D, making it unsuitable.

Suterland-Hodgeman Polygon Clipping

Four test cases:

1. First vertex inside and the second outside (in-out pair)
2. Both vertices inside clip window
3. First vertex outside and the second inside (out-in pair)
4. Both vertices outside the clip window

Concave polygons may be displayed with extra lines.



Weiler-Atherton Polygon Clipping

- Clips concave polygons correctly.
- Instead of always going around the polygon edges, we also, want to follow window boundaries.

1. For an outside-to-inside pair of vertices, follow the polygon boundary.
2. For an inside-to-outside pair of vertices, follow the window boundary in a clockwise direction.