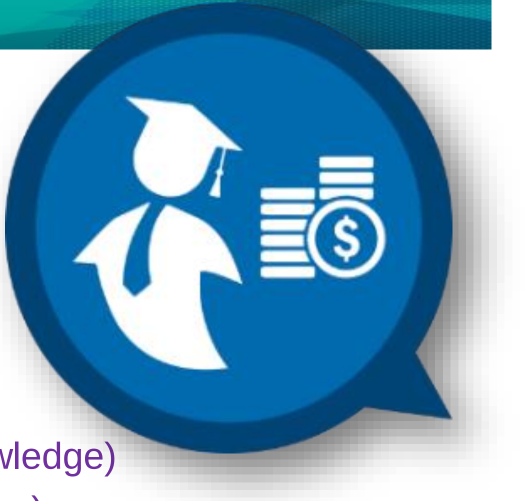
Universiti Malaysia PAHANG
Engineering • Technology • Creativity

Computer Graphics

2D Viewing and Clipping

Prepared by
Dr. Md. Manjur Ahmed
Faculty of Computer Systems and Software Engineering
manjur@ump.edu.my

Chapter Description



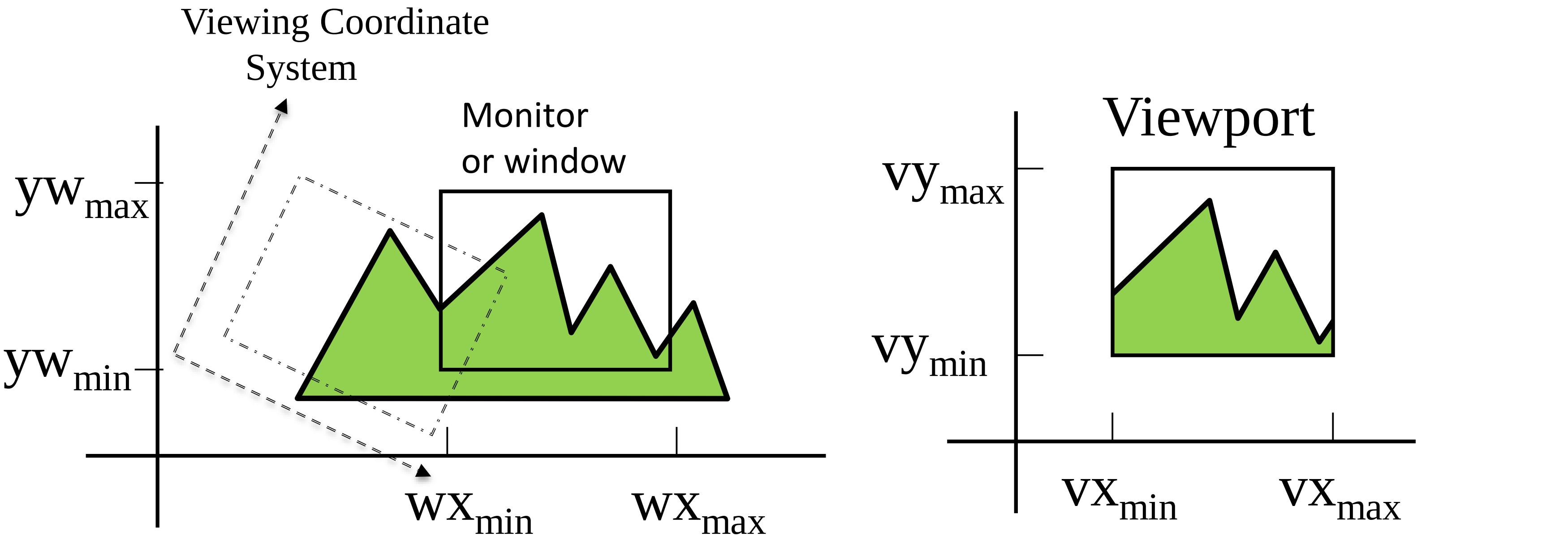
- Aims
  - Basic of Computer Graphics.

- Expected Outcomes
  - Understand the basic concept of computer graphics. (CO1: Knowledge)
  - Ability to use the computer graphics technology. (CO1: Knowledge)

- References
  - Computer Graphics by Zhigang Xiang, Schaum’s Outlines.
  - Donald Hearn & M. Pauline Baker, Computer Graphics with OpenGL, 4th Edition, Boston : Addison Wesley, 2011.

Viewing transformation

- Master coordinate system, commonly referred to as the world coordinate system
    - ✓ Clipping window: What do we want to see?
    - ✓ Viewport: Where do we want to see it?

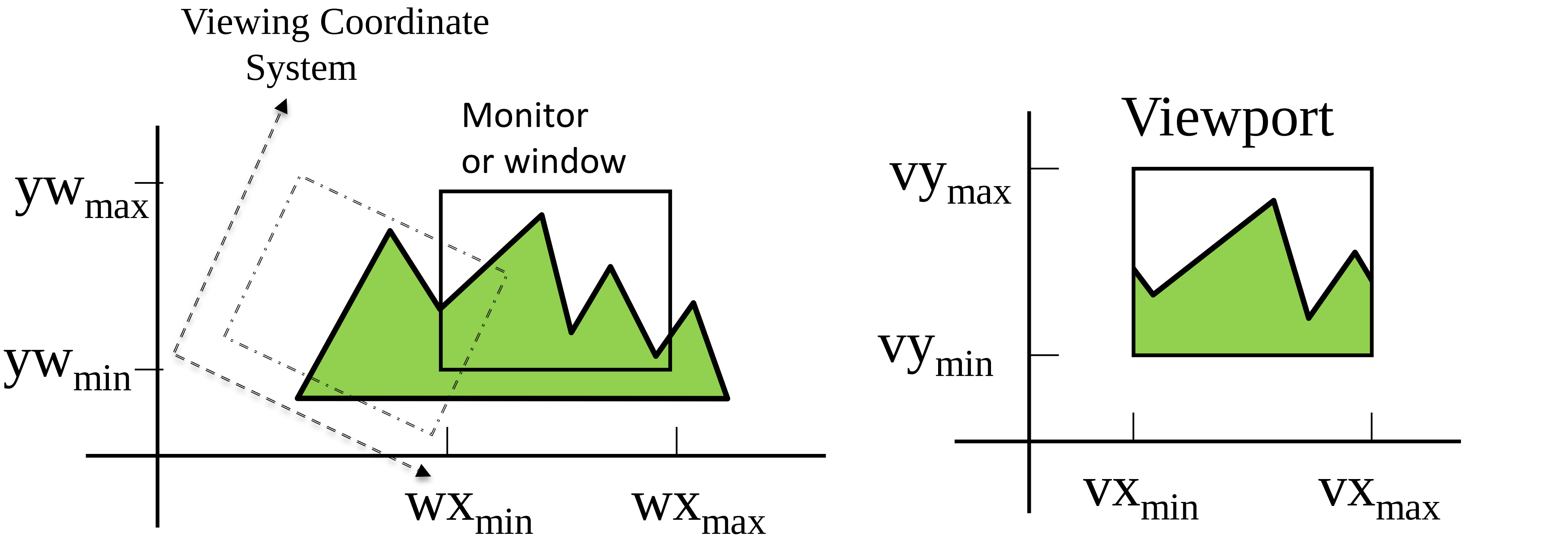


a) World Coordinate System

b) Device Coordinate System

Viewing transformation

- **Viewing Transformation**: coordinate-mapping operations between world and viewing coordinate system.
  - ✓ clipping window is mapped to the viewport



a) World Coordinate System

b) Device Coordinate System

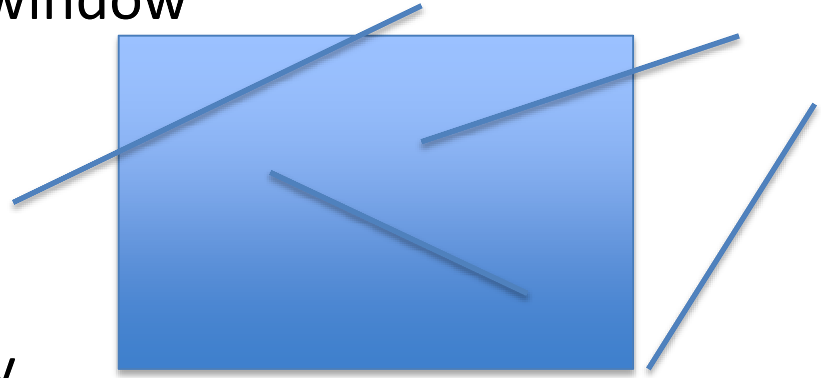
Clipping

- Objects in the scene possibly will be completely

  (a) inside the window,

  (b) outside the window,

  Or (c) partially visible through the window



- Why ?

  Because of reducing time complexity....

  - Reduce time complexity to avoid the scan converting pixels outside window
  - Therefore, avoid time and iteration because of rasterizing outside of framebuffer bounds

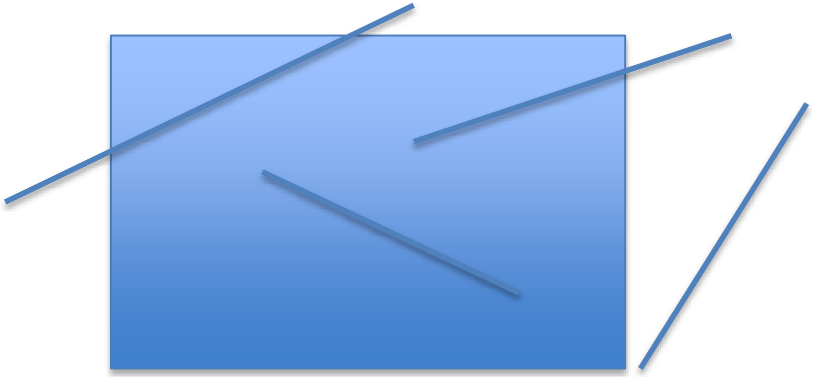
2D Clipping algorithms

- **Cohen-Sutherland Algorithm (Line)**
- **Liang-Barsky algorithm (Line)**
- **Sutherland-Hodgeman Algorithm (Polygon)**

Line Clipping

Possible Configuration:

1. Both endpoints are inside the region
   - No clipping necessary



2. One endpoint in, one out
   - Clip at intersection point

3. Both endpoints outside the region:
   a. No intersection
   b. Line intersects the region
      - Clip line at both intersection points

Line Clipping: Cohen-Sutherland Algorithm

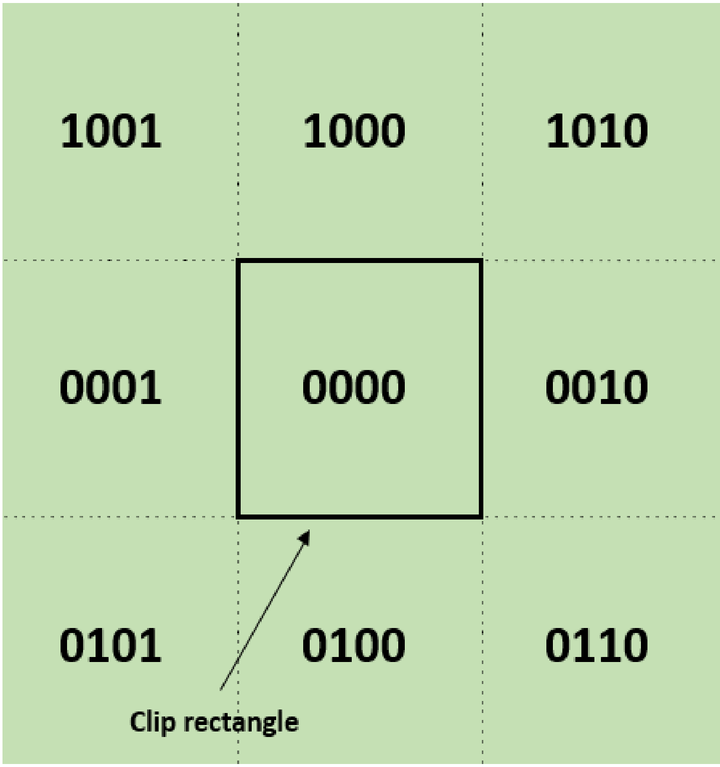
- Concept: Let a line with end point pairs (x_1,y_1) and (x_2,y_2)
  - Trivial Accept/Rejects
  - Initial tests on a line for acceptance or rejection:
    - Determine whether intersection calculations is required.
  - If neither be trivially accepted nor rejected, this line is divided into two segments at a clip edge. Thus, one segment can be trivially rejected.

Cohen-Sutherland Algorithm

- Assign a 4-bit region code to each endpoint c_0, c_1

Bit1 = 1 if $y > y_{max}$ i.e. 1000
Bit2 = 1 if $y < y_{min}$ i.e. 0100
Bit3 = 1 if $x > x_{max}$ i.e. 0010
Bit4 = 1 if $x < x_{min}$ i.e. 0001
========================

```
Algorithm: accept/reject
if ($c_0 \mid c_1$) = 0000
    accept (draw)
else if ($c_0$ & $c_1$) $\neq$ 0000
    reject (don't draw)
else clip using intersection points
    and retest
```



1001	1000	1010
0001	0000	0010
0101	0100	0110

Intersect point

- If 1000, intersect with line $y=y_{max}$
- If 0100, intersect with line $y=y_{min}$
- If 0010, intersect with line $x=x_{max}$
- If 0001, intersect with line $x=x_{min}$

Intersect Point (x_i, y_i)

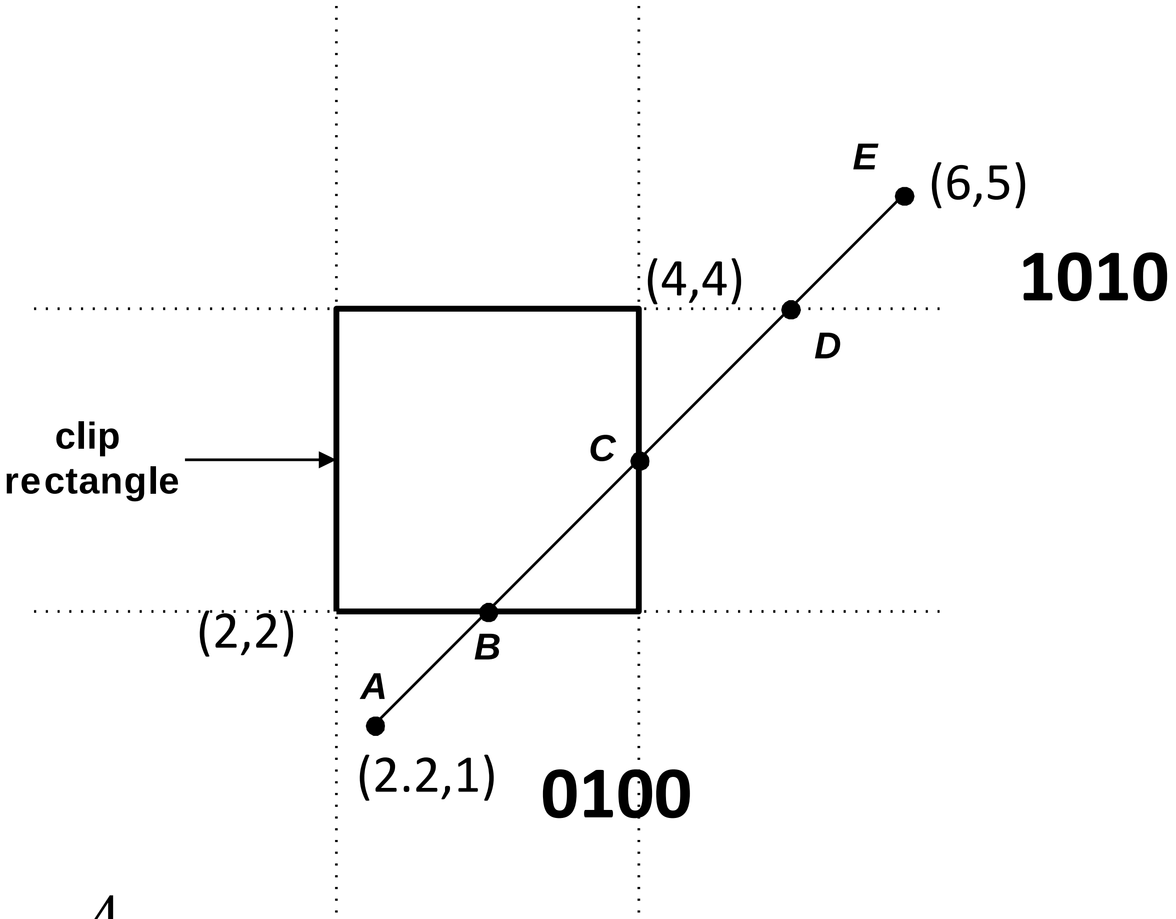
$$\left.\begin{aligned} x_i &= x_{\min} \text{ or } x_{\max} \\ y_i &= y_1 + m(x_i - x_1) \end{aligned}\right\}$$ If edge line is vertical

OR

$$\left.\begin{aligned} x_i &= x_1 + (y_i - y_1)/m \\ y_i &= y_{\min} \text{ or } y_{\max} \end{aligned}\right\}$$ If edge line is horizontal

$$\textit{where,}\quad m = (y_2 - y_1)/(x_2 - x_2)$$

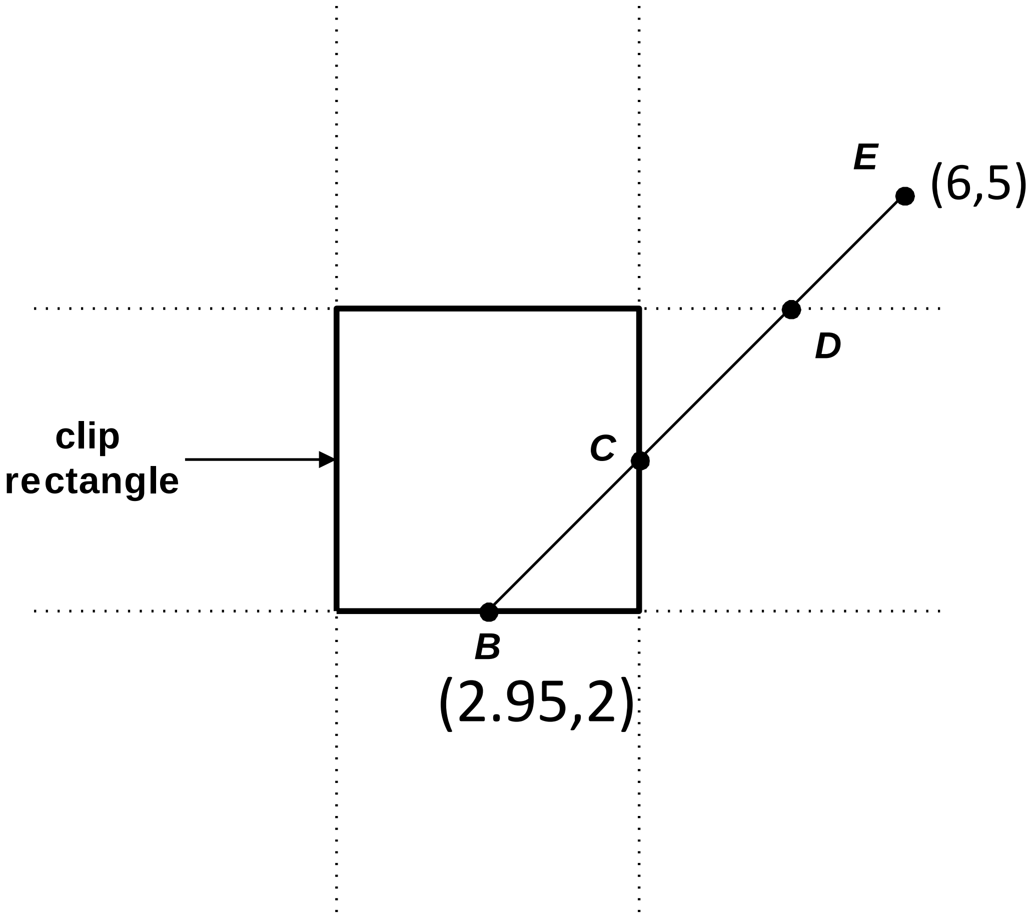
Cohen-Sutherland Algorithm



$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5-1}{6-2.2} = \frac{4}{3.8} = 1.05$$

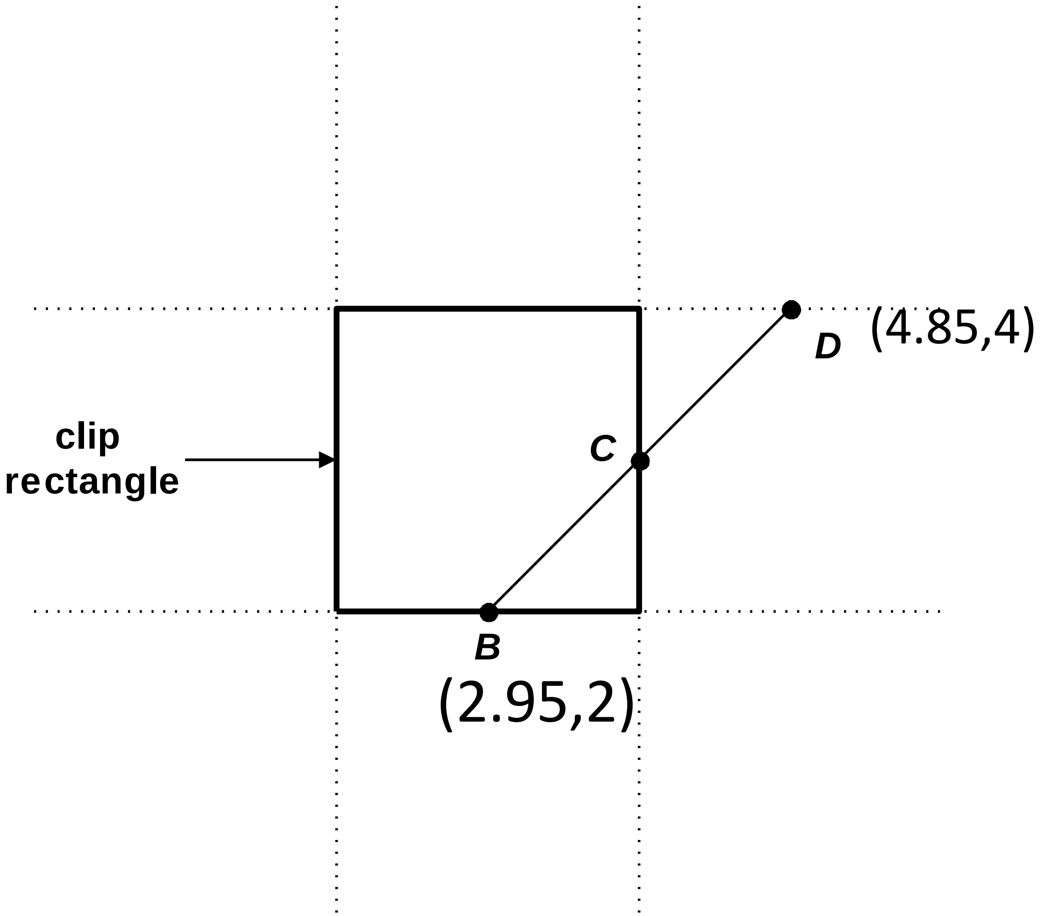
An Example

Cohen-Sutherland Algorithm



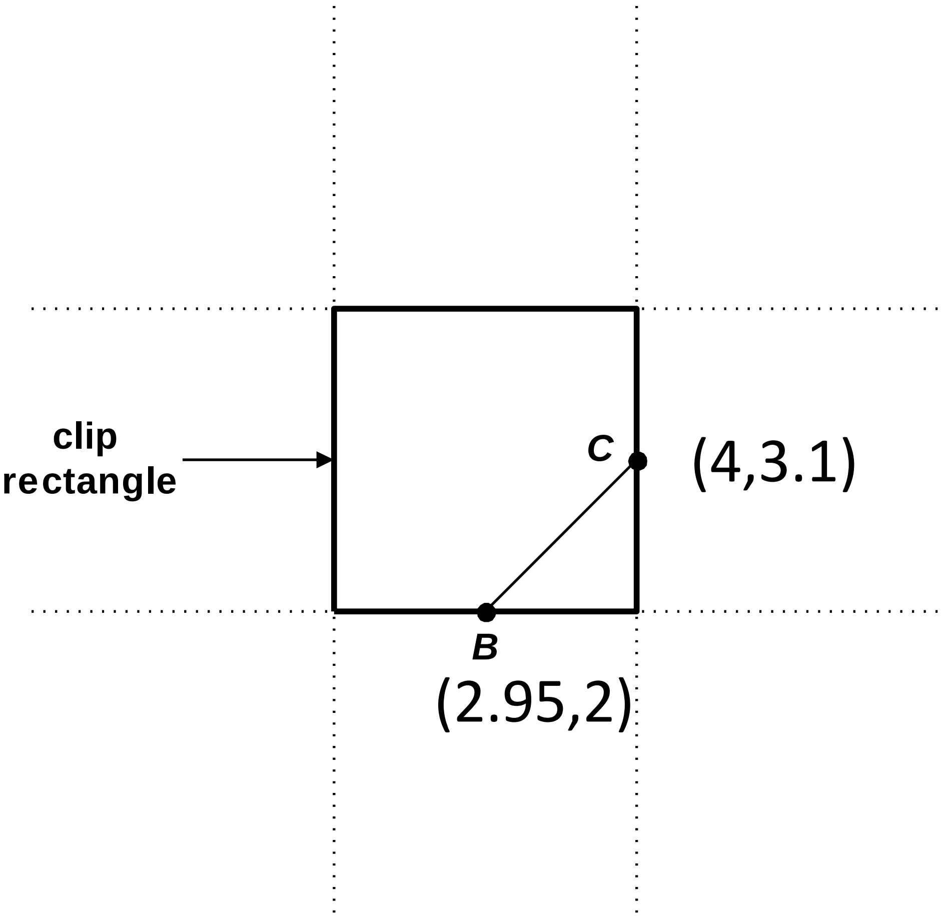
An Example

Cohen-Sutherland Algorithm



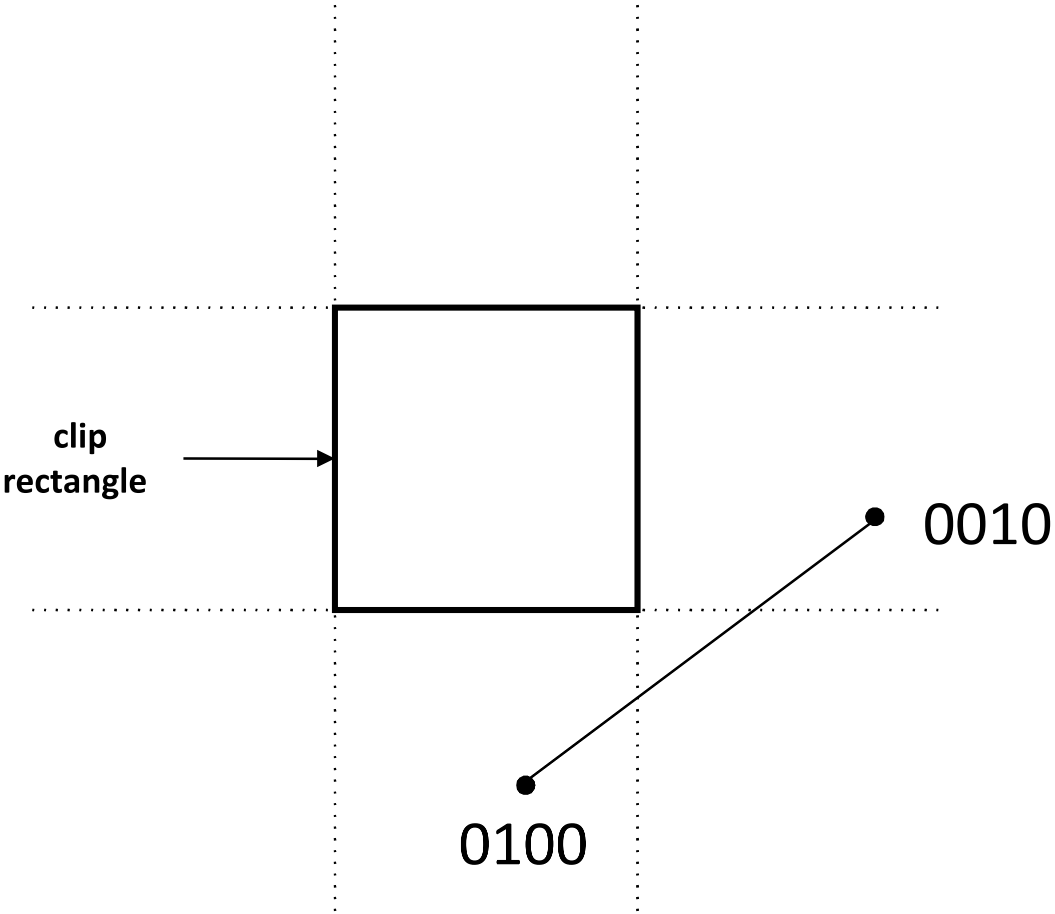
An Example

Cohen-Sutherland Algorithm

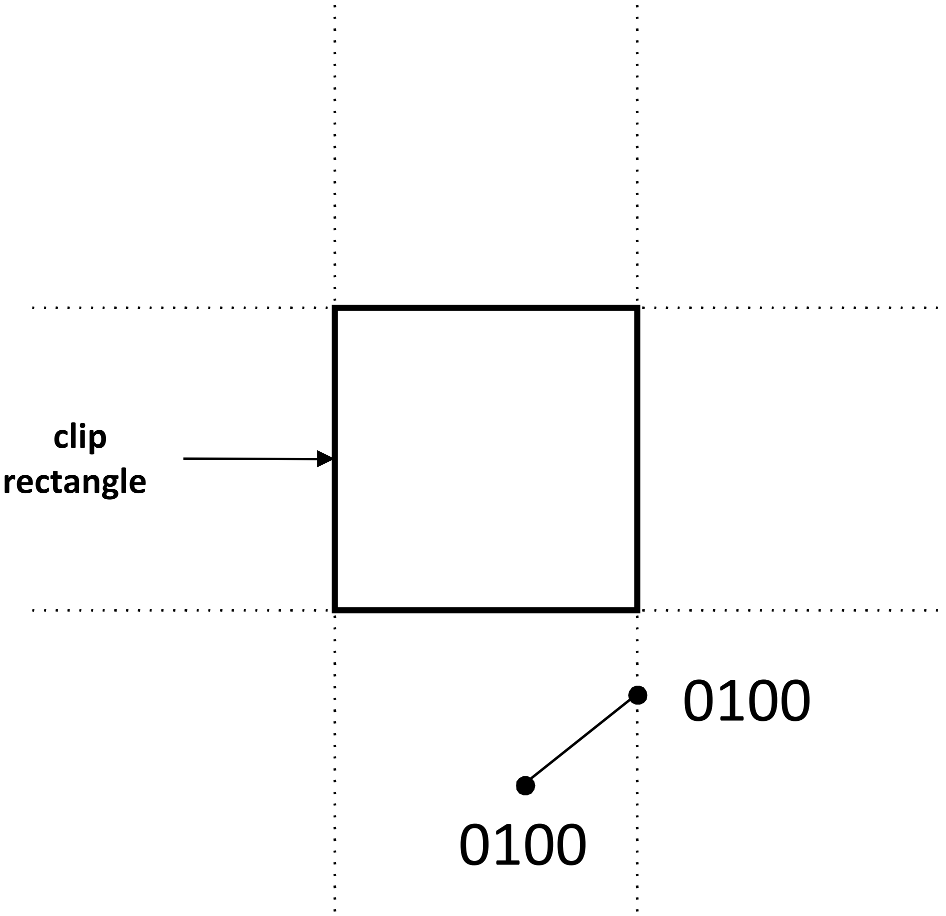


An Example

Another Example



Another Example


clip rectangle
0100
0100