UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

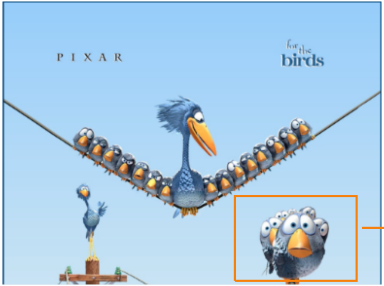
DEPARTAMENTO
DE INFORMÁTICA

Computación Gráfica – Clipping de segmentos

Versión 220317

Clipping de segmentos

De la clase anterior, sabemos que el proceso de *Viewing* nos permite determinar qué parte de nuestro “mundo” deseamos ver en nuestra pantalla. Esto se logra mediante la selección de una “window” en el “mundo”.



window

Objeto (mundo)

$\xrightarrow{f_n(x,y)}$

NDC

Clipping de segmentos

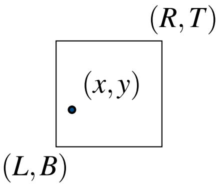
Clipping: técnica que separa los elementos de la figura en visibles y no–visibles.

Los algoritmos de clipping dependen de la forma de la “ventana”.

Para efectos del curso, estudiaremos el clipping en ventanas rectangulares.

Clipping 2D: para un punto.

Clipping sobre un punto es fácil de calcular.



El punto (x,y) es visible si:

$$L \leq x \leq R$$
$$B \leq y \leq T$$

Pero separa una figura en todos sus puntos es ineficiente.

Solución: aplicar algoritmos de intersección de líneas y polígonos.

Clipping 2D: para una línea.

Algoritmo:

1. Recorrer todos los segmentos de línea y seleccionar aquellos que intersectan la ventana.

Clipping 2D: para una línea.

Algoritmo:

1. Recorrer todos los segmentos de línea y seleccionar aquellos que intersectan la ventana.
2. Recortar los segmentos del paso 1, calculando su intersección con el borde.

Clipping 2D: para una línea.

Algoritmo:

1. Recorrer todos los segmentos de línea y seleccionar aquellos que intersectan la ventana.
2. Recortar los segmentos del paso 1, calculando su intersección con el borde.

El paso 1 puede separar los segmentos en:

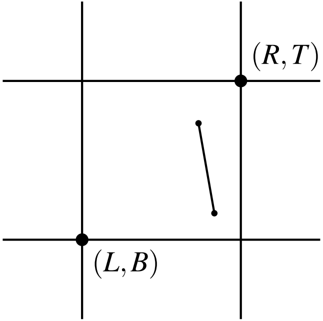
- Visibles: ambos extremos están dentro de la ventana.
- Invisibles: ambos extremos están fuera de la ventana*.
- Indeterminados: el resto.

*: Falta una condición.

Clipping 2D: para una línea.

La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$
$$x_1 > R \wedge x_2 > R$$
$$y_1 < B \wedge y_2 < B$$
$$y_1 > T \wedge y_2 > T$$



Visible:

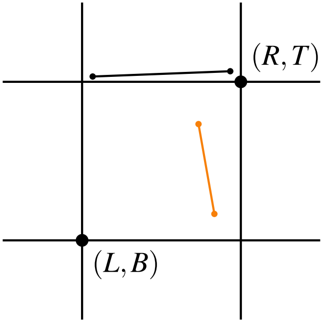
Invisible:

Indeterminado:

Clipping 2D: para una línea.

La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$
$$x_1 > R \wedge x_2 > R$$
$$y_1 < B \wedge y_2 < B$$
$$y_1 > T \wedge y_2 > T$$



Visible:

Invisible:

Indeterminado:

Clipping 2D: para una línea.

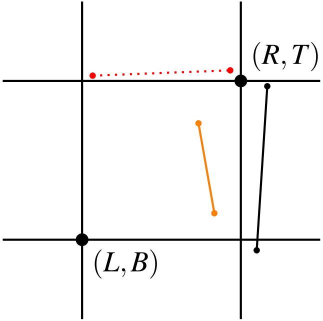
La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$

$$x_1 > R \wedge x_2 > R$$

$$y_1 < B \wedge y_2 < B$$

$$y_1 > T \wedge y_2 > T$$



Visible:

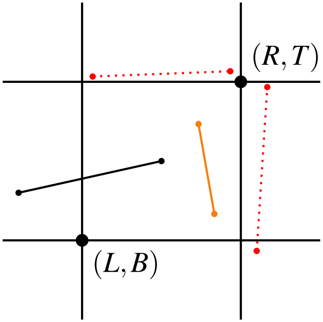
Invisible:

Indeterminado:

Clipping 2D: para una línea.

La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$
$$x_1 > R \wedge x_2 > R$$
$$y_1 < B \wedge y_2 < B$$
$$y_1 > T \wedge y_2 > T$$



Visible:

Invisible:

Indeterminado:

Clipping 2D: para una línea.

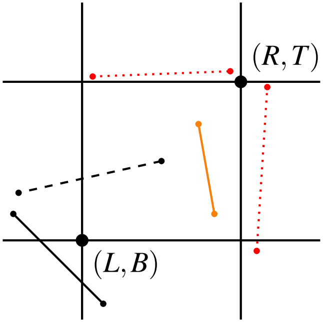
La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$

$$x_1 > R \wedge x_2 > R$$

$$y_1 < B \wedge y_2 < B$$

$$y_1 > T \wedge y_2 > T$$



Visible:

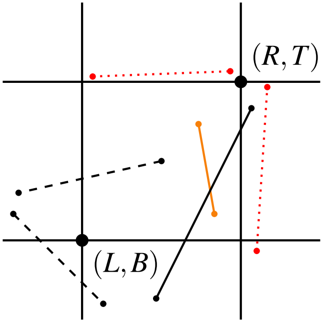
Invisible:

Indeterminado:

Clipping 2D: para una línea.

La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$
$$x_1 > R \wedge x_2 > R$$
$$y_1 < B \wedge y_2 < B$$
$$y_1 > T \wedge y_2 > T$$



Visible:

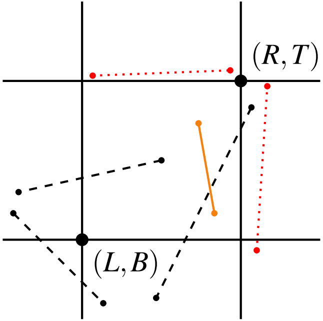
Invisible:

Indeterminado:

Clipping 2D: para una línea.

La condición faltante es, será invisible si se cumple alguna de las siguientes:

$$x_1 < L \wedge x_2 < L$$
$$x_1 > R \wedge x_2 > R$$
$$y_1 < B \wedge y_2 < B$$
$$y_1 > T \wedge y_2 > T$$



Visible:

Invisible:

Indeterminado:

Clipping 2D: Cohen-Sutherland.

Algoritmo de Cohen-Sutherland

Se asigna de un código de 4 bits a las nueve regiones:

- Bit 1: se enciende cuando estoy sobre la *window*.
- Bit 2: se enciende cuando estoy bajo la *window*.
- Bit 3: se enciende cuando estoy a la derecha la *window*.
- Bit 4: se enciende cuando estoy a la izquierda la *window*.

1001	0001	0101
1000	0000	0100
1010	0010	0110

Clipping 2D: Cohen-Sutherland.

Algoritmo de Cohen-Sutherland: analizar los bits de los nodos extremos del segmento.

- Si el mismo *bit* está encendido en ambos extremos, el segmento es invisible. Ejemplo: (1010) y (0010).
- Si ambos extremos tienen código (0000), el segmento está adentro de la *window* y es visible.
- En cualquier otro caso es indeterminado.

1001	0001	0101
1000	0000	0100
1010	0010	0110

Clipping 2D: Cohen-Sutherland.

Pseudo–algoritmo:

if $codigoP1 == 0\&\&codigoP2 == 0$ **then**
  /* es y lógico */
  Segmento visible
else
  if $codigoP1\&codigoP2 \neq 0$ **then**
    /* es y binario */
    Segmento no–visible
  else
    Indeterminado
  end if
end if

1001	0001	0101
1000	0000	0100
1010	0010	0110

Cálculo de las intersecciones

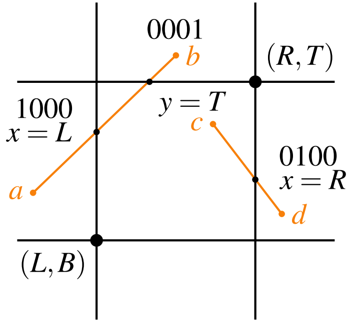
Para los segmentos indeterminados, hay que calcular la intersección del segmento con el borde de la *window*: punto(s) de intersección entre las ecuaciones del segmento y del borde.

- ¿Es necesario calcular la intersección del segmento con cada lado de la window?
- ¿Cuántos lados de la window rectangular pueden ser interceptados por un segmento?

Clipping 2D: para una línea.

Análisis de bits en la intersección

- Bit1 encendido $\Rightarrow$ intersección en $y = T$.
- Bit2 encendido $\Rightarrow$ intersección en $y = B$.
- Bit3 encendido $\Rightarrow$ intersección en $x = R$.
- Bit4 encendido $\Rightarrow$ intersección en $x = L$.



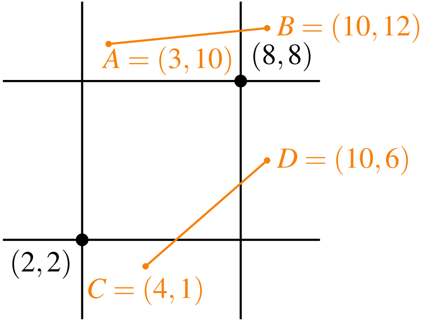
Clipping 2D para una línea: Ejemplo

Ecuación paramétrica de la recta:

Si $0 \leq t \leq 1$ entonces el punto está dentro del segmento.

$$x = x_1 + t \cdot (x_2 - x_1)$$

$$y = y_1 + t \cdot (y_2 - y_1)$$



1001	0001	0101
1000	0000	0100
1010	0010	0110

Clipping 2D para una línea: Ejemplo

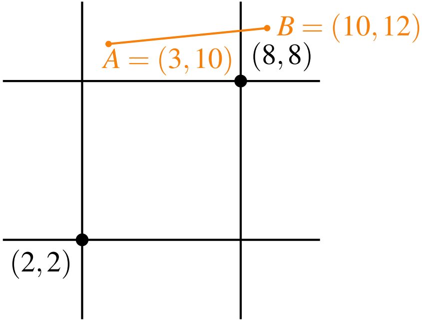
Solución para el segmento AB:

código de $A = (0001)$.

código de $B = (0101)$.

$(0001\&0101) = (0001) \neq 0$,

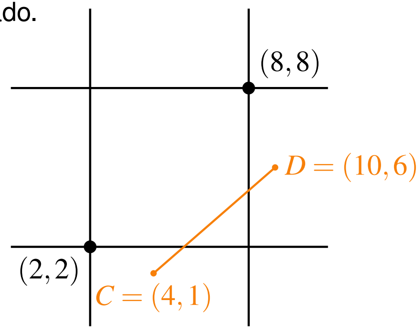
entonces AB no es visible.



1001	0001	0101
1000	0000	0100
1010	0010	0110

Clipping 2D para una línea: Ejemplo

Solución para el segmento CD:
código de $C = (0010)$.
código de $D = (0100)$.
$(0010\&0100) = 0$,
entonces CD es indeterminado.



1001	0001	0101
1000	0000	0100
1010	0010	0110

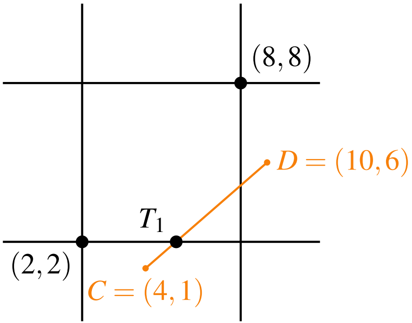
Clipping 2D para una línea: Ejemplo

Solución para el segmento CD:

$C = (4,1) \Rightarrow$ código $(0010) \Rightarrow 2^{do}$ bit intersecta en $B = 2$.
$x = 4 + t \cdot (10 - 4) = 4 + 6t$
$y = 1 + t \cdot (6 - 1) = 1 + 5t$

Reemplazando en $y = 2$:
$2 = 1 + 5t \Rightarrow t = \frac{1}{5}$
$x = 4 + \frac{6}{5} \Rightarrow x = \frac{26}{5}$

Punto de intersección:
$T_1 = (\frac{26}{5}, 2)$.



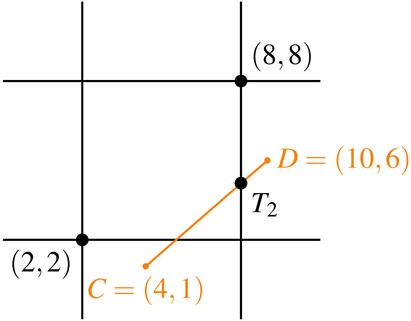
Clipping 2D para una línea: Ejemplo

Solución para el segmento CD:

$D = (10,6) \Rightarrow$ código $(0100) \Rightarrow 3^{er}$ bit intersecta en $R = 8$.
$x = 4 + t \cdot (10 - 4) = 4 + 6t$
$y = 1 + t \cdot (6 - 1) = 1 + 5t$

Reemplazando en $x = 8$:
$8 = 4 + 6t \Rightarrow t = \frac{2}{3}$
$y = 1 + \frac{10}{3} \Rightarrow y = \frac{13}{3}$

Punto de intersección:
$T_2 = (8, \frac{13}{3})$.



Clipping 2D Liang–Barsky

Recordemos que la zona de clipping está limitada por:

$$L \leq x \leq R$$
$$B \leq y \leq T$$

Además:

$$x = x_1 + t \cdot (x_2 - x_1) \qquad donde \quad 0 \leq t \leq 1$$
$$y = y_1 + t \cdot (y_2 - y_1)$$

Definiendo:

$$d_x = x_2 - x_1$$
$$d_y = y_2 - y_1$$

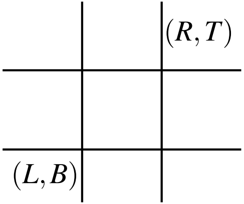
Clipping 2D Liang–Barsky

Definiendo:

$$
\begin{aligned}
d_x &= x_2 - x_1 & d_y &= y_2 - y_1 \\
d_1 &= -d_x & d_3 &= -d_y \\
d_2 &= d_x & d_4 &= dy
\end{aligned}
$$

Además:

$$
\begin{aligned}
q_1 &= x_1 - L & q_3 &= y_1 - B \\
q_2 &= R - x_1 & q_4 &= T - y_1
\end{aligned}
$$



Primer resultado:

- Si $d_1 = d_2 = 0 \Rightarrow$ segmento paralelo al eje Y.
- Si $d_3 = d_4 = 0 \Rightarrow$ segmento paralelo al eje X.

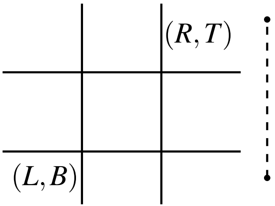
Clipping 2D Liang–Barsky

Definiendo:

$$d_x = x_2 - x_1 \qquad d_y = y_2 - y_1$$
$$d_1 = -d_x \qquad d_3 = -d_y$$
$$d_2 = d_x \qquad d_4 = dy$$

Además:

$$q_1 = x_1 - L \qquad q_3 = y_1 - B$$
$$q_2 = R - x_1 \qquad q_4 = T - y_1$$



$d_2 = 0 \wedge q_2 < 0$

Entonces:

- Si $d_1 = 0 \wedge q_1 < 0 \Rightarrow$ no visible.
- Si $d_2 = 0 \wedge q_2 < 0 \Rightarrow$ no visible.
- Si $d_3 = 0 \wedge q_3 < 0 \Rightarrow$ no visible.
- Si $d_4 = 0 \wedge q_4 < 0 \Rightarrow$ no visible.

Caso general

Recordando que: $x = x_1 + t \cdot (x_2 - x_1)$ donde $t \in [0, 1]$.

Reemplazando en punto del borde $x = L$:
$L = x_1 + t \cdot d_x$, reordenando:
$t = \frac{(L - x_1)}{d_x}$, pero $(x_1 - L) = q_1$ y $d_1 = -d_x$, entonces:
$t = \frac{q_1}{d_1}$.

Reemplazando en punto del borde $x = R$:
$R = x_1 + t \cdot d_x$, reordenando:
$t = \frac{(R - x_1)}{d_x}$, pero $(R - x_1) = q_2$ y $d_2 = d_x$, entonces:
$t = \frac{q_2}{d_2}$.

Caso general

Recordando que: $y = y_1 + t \cdot (y_2 - y_1)$ donde $t \in [0,1]$.

Reemplazando en punto del borde $y = B$:
$B = y_1 + t \cdot d_y$, reordenando:
$t = \frac{(B - y_1)}{d_y}$, pero $(y_1 - B) = q_3$ y $d_3 = -d_y$, entonces:
$t = \frac{q_3}{d_3}$.

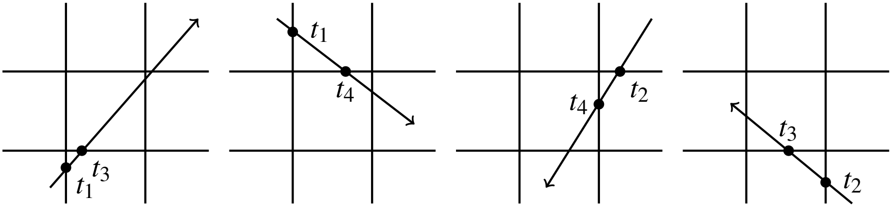
Reemplazando en punto del borde $y = T$:
$T = y_1 + t \cdot d_y$, reordenando:
$t = \frac{(T - y_1)}{d_y}$, pero $(T - y_1) = q_4$ y $d_4 = d_y$, entonces:
$t = \frac{q_4}{d_4}$.

Resumiendo

- Le intersección con los límites de la zona de clipping ocurren $t_i = \frac{q_i}{d_i}$ con $i = 1, \ldots, 4$.
- Falta determinar dos valores de t que definen la intersección del segmento con la zona de clipping.
- Llamemos a estos límites t_a y t_b.

Clipping 2D Liang–Barsky

Analizando los casos para $d_i < 0$.



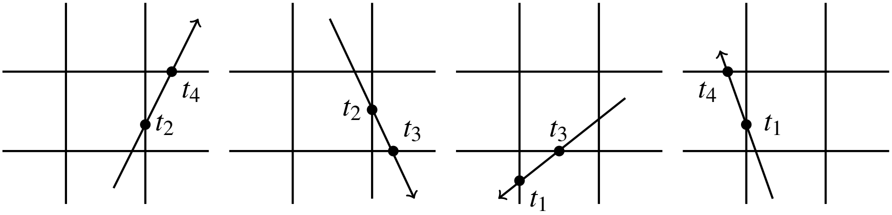
$d_1 < 0 \wedge d_3 < 0$ $d_1 < 0 \wedge d_4 < 0$ $d_2 < 0 \wedge d_4 < 0$ $d_2 < 0 \wedge d_3 < 0$

$$t_a = \text{máx}\{t_i = \frac{q_i}{d_i} | d_i < 0\} (*)$$

(*): falta una condición.

Clipping 2D Liang–Barsky

Analizando los casos para $d_i > 0$.



$d_2 > 0 \wedge d_4 > 0$ $\quad$ $d_2 > 0 \wedge d_3 > 0$ $\quad$ $d_1 > 0 \wedge d_3 > 0$ $\quad$ $d_1 > 0 \wedge d_4 > 0$

$$t_b = \text{mín}\{t_i = \frac{q_i}{d_i} | d_i > 0\}(*)$$

(*): falta una condición.

Clipping 2D Liang–Barsky

La condición que falta se establece del hecho que $t \in [0, 1]$, luego:

- t_a está acotado por 0, i.e., el punto inicial de la sección visible está al interior de la ventana de clipping.
- t_b está acotado por 1, i.e., el punto final de la sección visible está al interior de la ventana de clipping.

Entonces:

$$t_a = \max\{t_i = \frac{q_i}{d_i} | d_i < 0, 0\}$$

$$t_b = \min\{t_i = \frac{q_i}{d_i} | d_i > 0, 1\}$$

Luego, si $t_a > t_b \Rightarrow$ el segmento no es visible. En caso contrario, la sección visible está definida por $t_a \leq t \leq t_b$.

Propuesto: programar el algoritmo.

Clipping 2D: División en punto medio

- Un segmento que no es completamente visible se divide en el punto medio.
- Se descarta el lado no visible.
- Si el otro lado aún no es completamente visible, se repite el proceso.

$$P_1 \leftarrow P$$

$$P \leftarrow \frac{P_1 + P_2}{2}$$