Sección de Informática Gráfica | Computer Graphics Group VALENCIA

UNIVERSIDAD POLITECNICA DE VALENCIA

Transformaciones y Visualización 2D

Fundamentos matemáticos

Transformaciones 2D

Visualización 2D

Recortado 2D

Fundamentos

- La geometría es el estudio de las relaciones entre objetos en un espacio n-dimensional
  - En gráficos, nos interesan los espacios de 2 y 3 dimensiones
- Necesitamos un conjunto mínimo de primitivas con las que construir objetos más complejos:
  - Escalares
  - Vectores
  - Puntos

Fundamentos

- La geometría proporciona los fundamentos matemáticos necesarios en Informática gráfica:
  - Espacios geométricos: Vectorial, Afín, Cartesiano,…
  - Geometría Afín
  - Transformaciones afines
  - Perspectiva
  - Proyecciones
  - Representación matricial de proyecciones
  - Transformaciones de la vista

Fundamentos

- **Puntos:** se asocian a ubicaciones en el espacio
- **Vectores:** representan desplazamientos entre puntos o direcciones
- Los puntos, vectores y las operaciones que se realizan entre ellos, permiten resolver la mayoría de problemas en Modelado, Informática Gráfica, Animación, Visualización y Geometría computacional

Fundamentos

- Las dos operaciones principales con vectores son el producto escalar y el vectorial
- Aunque los puntos y los vectores se representen del mismo modo, son diferentes y no se deben mezclar
- Terminología:
  - Normal significa perpendicular, es decir que forma un ángulo de 90º
  - La norma o módulo de un vector es su tamaño
  - Ser coplanares, significa que un plano los contiene

Fundamentos

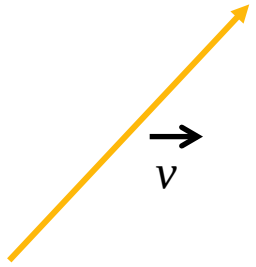
- La geometría se compone de tres elementos básicos: Escalares, Vectores y Puntos
- Escalares:
  - Se definen como miembros de conjuntos que se pueden combinar mediante sumas y productos
  - Estas operaciones tienen las propiedades asociativa, conmutativa e inversa.
  - Números reales, enteros, complejos…
  - Por sí solos no tienen propiedades geométricas

Fundamentos

- Los Vectores se utilizan para representar:
  - Puntos en el espacio (es decir, localizaciones)
  - Desplazamientos entre puntos
  - Direcciones (es decir, orientaciones)
- Los Puntos y los Vectores actúan de forma diferente:
  - Trasladar algo, significa moverlo sin cambiar su orientación
  - Trasladar un punto nos da como resultado un punto diferente
  - Al trasladar una dirección se mantiene la misma dirección

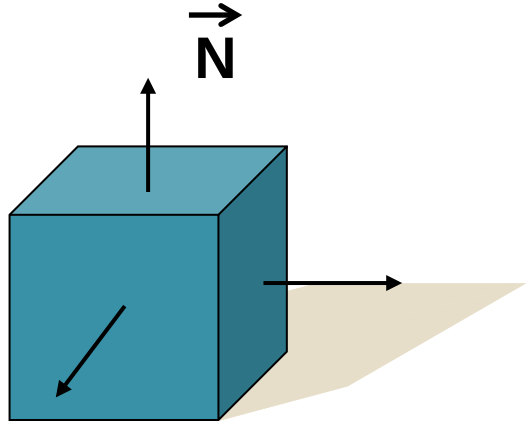
Fundamentos

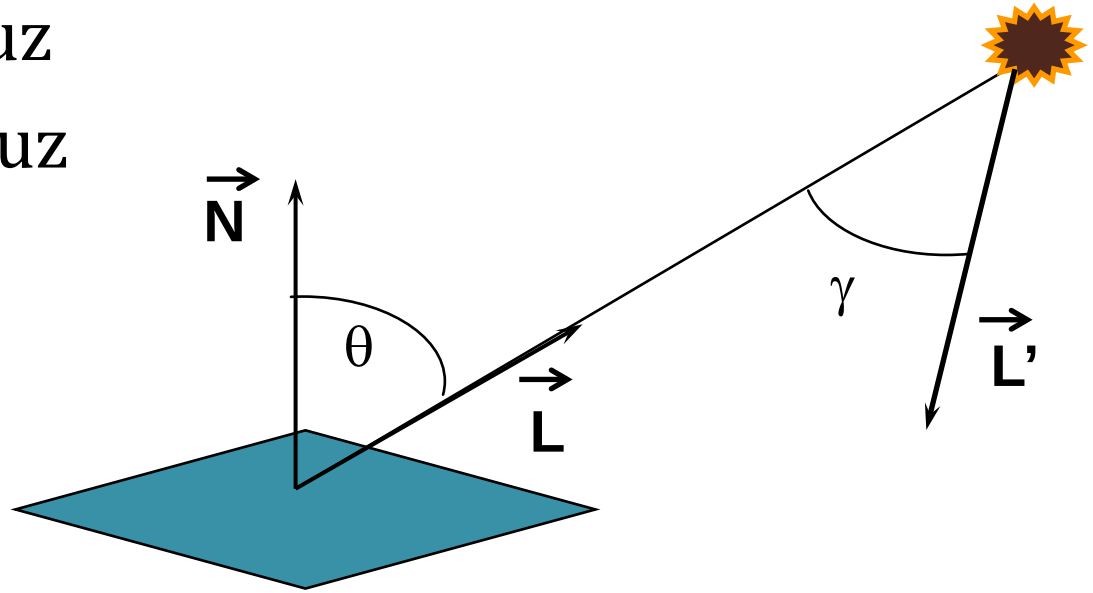
- La definición física de un Vector es una cantidad que tiene: dirección, sentido y magnitud
- Ejemplos de vectores:
  - Fuerza
  - Velocidad
  - Segmentos de recta dirigidos



Fundamentos

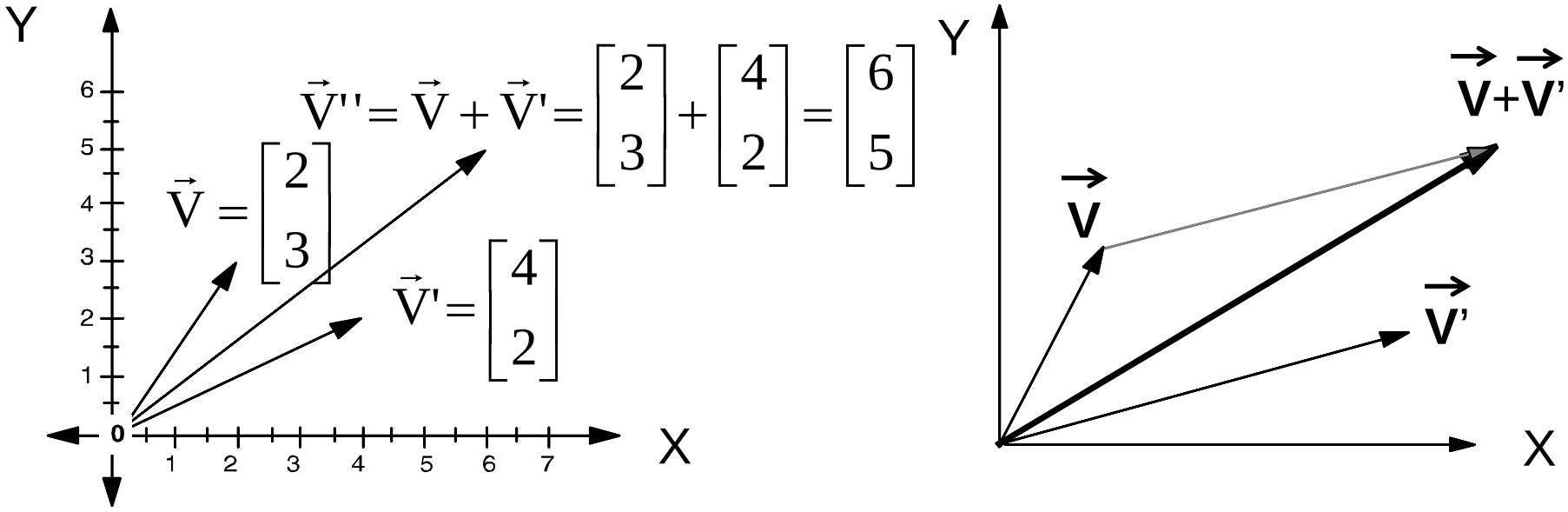
- **Los vectores se utilizan en gráficos para:**
  - Representar las posiciones de los vértices
  - Determinar la orientación de una superficie
    - Vector normal a la superficie
  - Modelar la interacción de la luz
    - Vector de incidencia de la luz





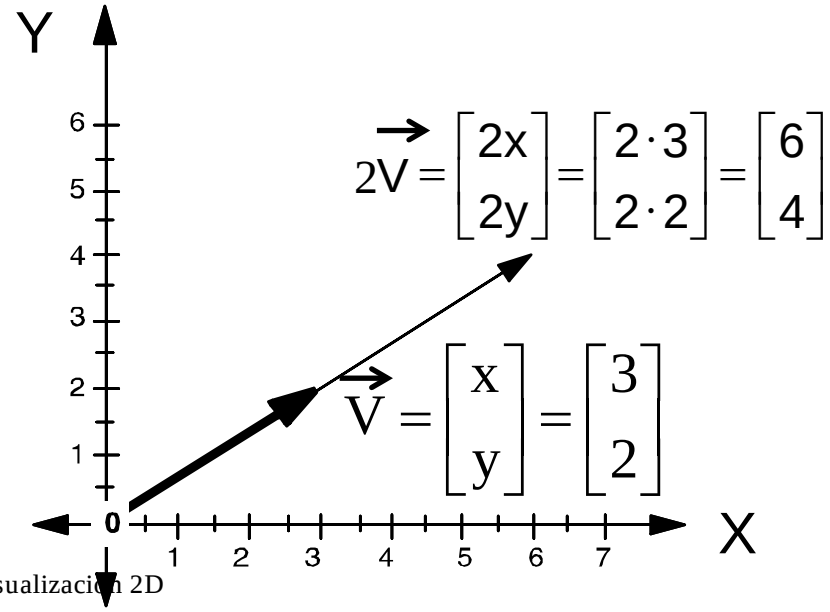
Fundamentos

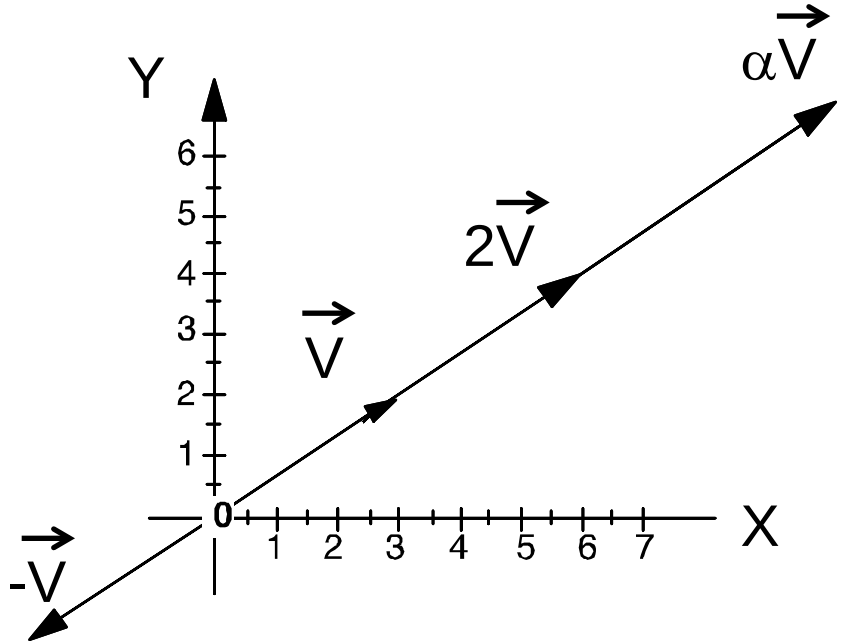
- Suma de vectores



Fundamentos

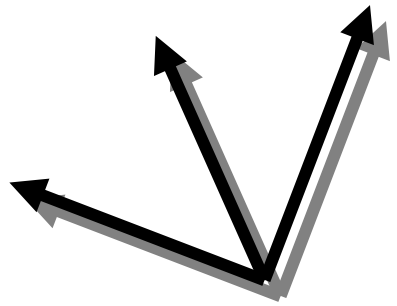
- Producto por un escalar
- Inverso
- Dependencia lineal
  - Dos vectores son linealmente dependientes cuando uno es un múltiplo de otro





Fundamentos

- Propiedades algebraicas de los vectores:
  - Conmutativa
    - $\vec{\mathbf{U}} + \vec{\mathbf{V}} = \vec{\mathbf{V}} + \vec{\mathbf{U}}$
  - Asociativa
    - $\vec{\mathbf{U}} + (\vec{\mathbf{V}} + \vec{\mathbf{R}}) = (\vec{\mathbf{U}} + \vec{\mathbf{V}}) + \vec{\mathbf{R}}$
  - Identidad (suma)
    - Existe un vector $\vec{\mathbf{0}}$ / $(\vec{\mathbf{V}} + \vec{\mathbf{0}}) = (\vec{\mathbf{0}} + \vec{\mathbf{V}}) = \vec{\mathbf{V}}$ para todo $\vec{\mathbf{V}}$
  - Inverso
    - Para cualquier $\vec{\mathbf{V}}$ existe un vector $-\vec{\mathbf{V}}$ / $\vec{\mathbf{V}} + (-\vec{\mathbf{V}}) = \vec{\mathbf{0}}$
  - Distributiva (suma vector)
    - $r\,(\vec{\mathbf{V}} + \vec{\mathbf{U}}) = r\,\vec{\mathbf{V}} + r\,\vec{\mathbf{U}}$
  - Distributiva (suma escalar)
    - $(r + s)\,\vec{\mathbf{V}} = r\,\vec{\mathbf{V}} + s\,\vec{\mathbf{V}}$
  - Asociativa (producto escalar)
    - $r(s\,\vec{\mathbf{V}}) = (rs)\,\vec{\mathbf{V}}$
  - Identidad (producto)
    - Para el número real 1, $1\,\vec{\mathbf{V}} = \vec{\mathbf{V}}$ para todo $\vec{\mathbf{V}}$



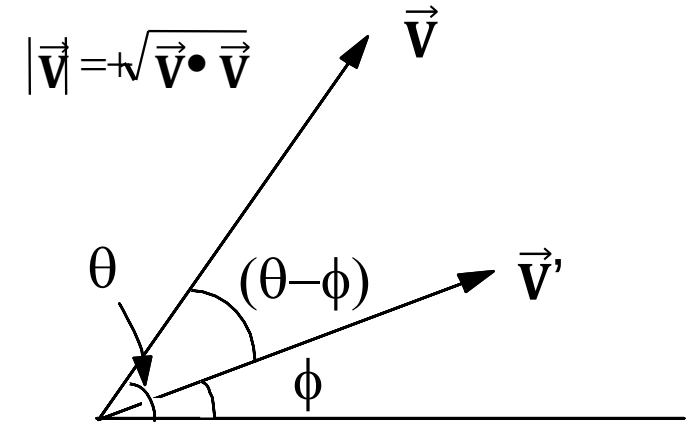
Fundamentos

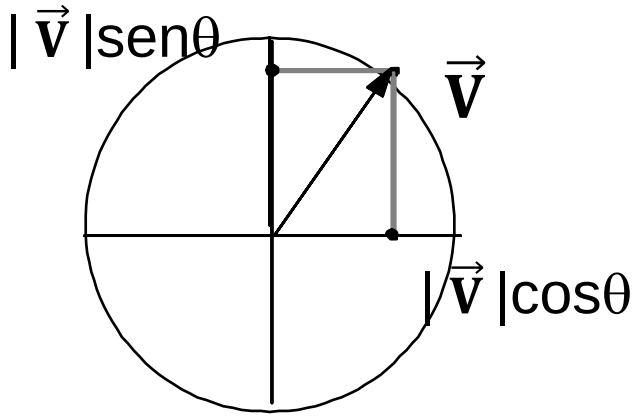
- Producto escalar de vectores
  - El resultado es escalar
  - Define la longitud de un vector

    $(\vec{V} * \vec{V}) = x^2 + y^2$ es el cuadrado de la longitud
  - Permite normalizar vectores
    - La norma de un vector es su longitud
    - Un vector es unitario si $|\vec{V}| = 1$
    - Para normalizar: $\vec{V} / |\vec{V}|$
  - Mide los ángulos entre dos vectores
    - Si $\vec{V}$ y $\vec{V}'$ son distintos de 0 entonces

      $\vec{V} * \vec{V}' = |\vec{V}||\vec{V}'|\cos(\theta-\phi)$
    - Si $\vec{V} * \vec{V}' > 0$ entonces $(\theta-\phi) < 90º$
    - Si $\vec{V} * \vec{V}' < 0$ entonces $(\theta-\phi) > 90º$

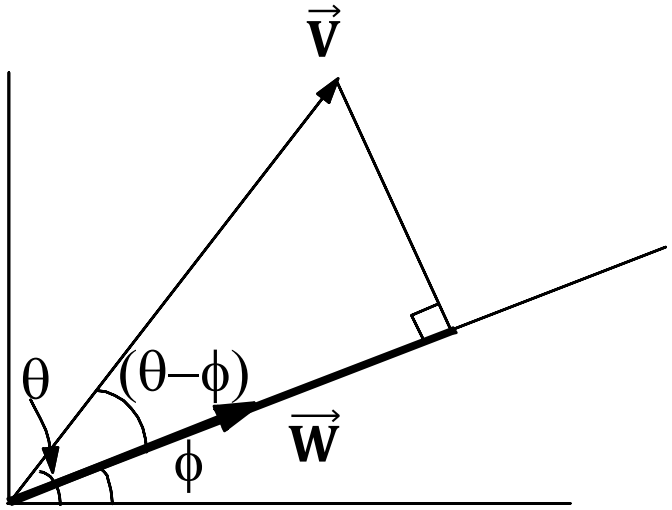
$$\vec{V} \bullet \vec{V} = \begin{bmatrix} x \\ y \end{bmatrix} \bullet \begin{bmatrix} x' \\ y' \end{bmatrix} = x \cdot x' + y \cdot y'$$



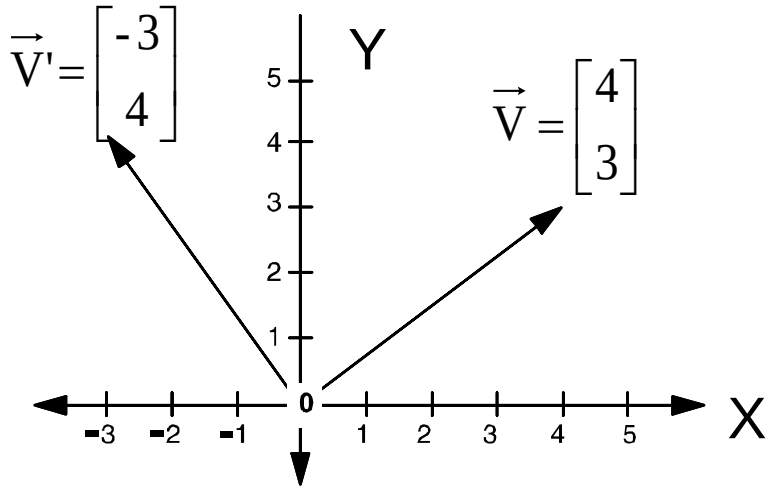


Fundamentos

- Permite calcular la longitud de la proyección de un vector sobre un eje
  - Si $\vec{W}$ es un vector unitario, entonces $\vec{V} * \vec{W}$ es la longitud de la proyección de $\vec{V}$ sobre la línea que contiene $\vec{W}$



- Determina si dos vectores son perpendiculares
  - El producto escalar de dos vectores perpendiculares es 0



$$\vec{V} \bullet \vec{V}' = x \cdot x' + y \cdot y' = (4 \cdot -3) + (3 \cdot 4) = 0$$

$$\vec{V} \bullet \vec{V}' = \left|\vec{V}\right| \cdot \left|\vec{V}'\right| \cdot \cos(90^o) = \left|\vec{V}\right| \cdot \left|\vec{V}'\right| \cdot 0 = 0$$

Fundamentos

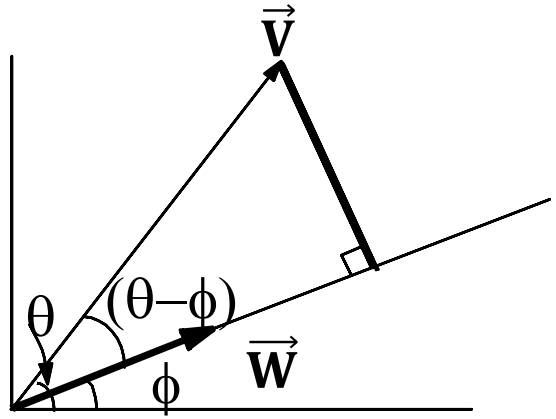
- Producto vectorial
  - El resultado es un vector (ortogonal a los vectores operando)
  - Mide los ángulos entre dos vectores
    - Si $\vec{V}$ y $\vec{V}'$ son distintos de $\vec{0}$ entonces

    $|\vec{V} \times \vec{V}'| = |\vec{V}||\vec{V}'|\text{sen}(\theta - \phi)$
  - Permite calcular la distancia de un punto a una recta
    - Si $\vec{W}$ es un vector unitario, entonces

    $|\vec{V} \times \vec{W}|$ es la distancia de $\vec{V}$ a la recta que contiene $\vec{W}$
  - Determina si dos vectores son paralelos
    - El módulo del producto vectorial de dos vectores paralelos es $\vec{0}$

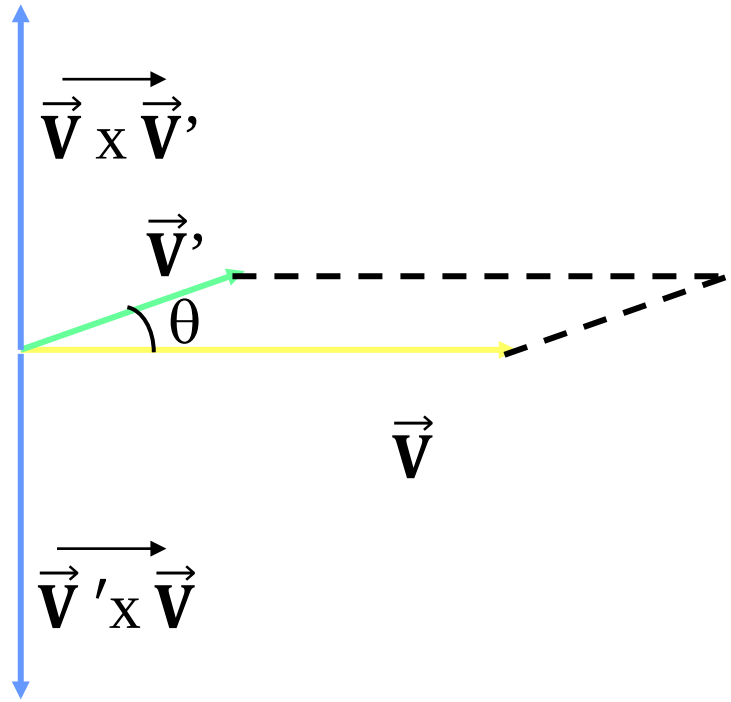
$$\vec{V} \times \vec{V}' = \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ x & y & z \\ x' & y' & z' \end{bmatrix}$$

$$\vec{V} \times \vec{V}' = \begin{bmatrix} yz' - zy' & zx' - xz' & xy' - yx' \end{bmatrix}$$



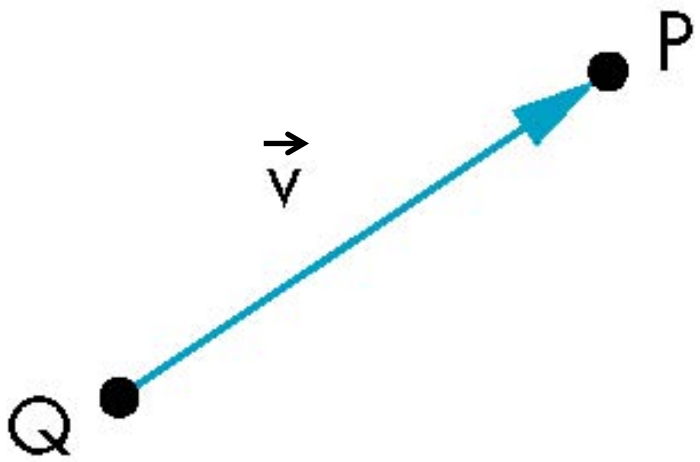
Fundamentos

- Producto vectorial
  - El módulo de $\vec{V}$ x $\vec{V}$' es igual al área del cuadrilátero definido por $\vec{V}$ y $\vec{V}$'.
  - El orden de los vectores es importante: $\vec{V} \times \vec{V}' = - \vec{V}' \times \vec{V}$
  - Propiedades:
    - $\vec{V} \times \vec{V}' = -(\vec{V}' \times \vec{V})$
    - $\vec{V} \times (\vec{V}' + \vec{W}) = (\vec{V} \times \vec{V}') + (\vec{V} \times \vec{W})$
    - $(\alpha \vec{V}) \times \vec{V}' = \vec{V} \times (\alpha \vec{V}') = \alpha(\vec{V} \times \vec{V}')$
    - $\vec{V} \times \vec{V} = \vec{0}$



Fundamentos

- Los Puntos definen una posición en el espacio
- Operaciones entre puntos y vectores:
  - Si se restan dos puntos se obtiene un vector
  - Equivalente a sumar un punto a un vector

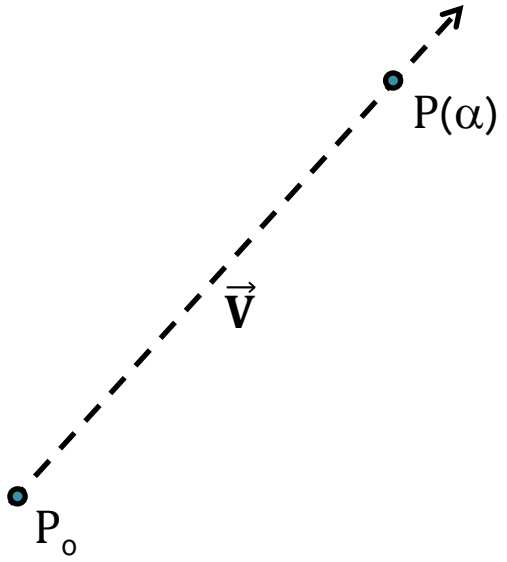


$$\vec{\mathbf{V}} = P - Q$$

$$P = \vec{\boldsymbol{V}} + Q$$

Fundamentos

- Una recta se define por todos los puntos:
  - $P(\alpha)=P_0 + \alpha \vec{\mathbf{V}}$
  - Conjunto de todos los puntos que pasan por P_0 en la dirección del vector $\vec{\mathbf{V}}$



Fundamentos

- Esta forma de definir la recta se denomina paramétrica
  - Más robusta y general que otras
  - Se puede aplicar a curvas y superficies
- Otras formas:
  - Explícito: $y = mx + b$
  - Implícito: $ax + by + c = 0$
  - Paramétrico:

$$x(\alpha) = \alpha x_0 + (1-\alpha)x_1$$

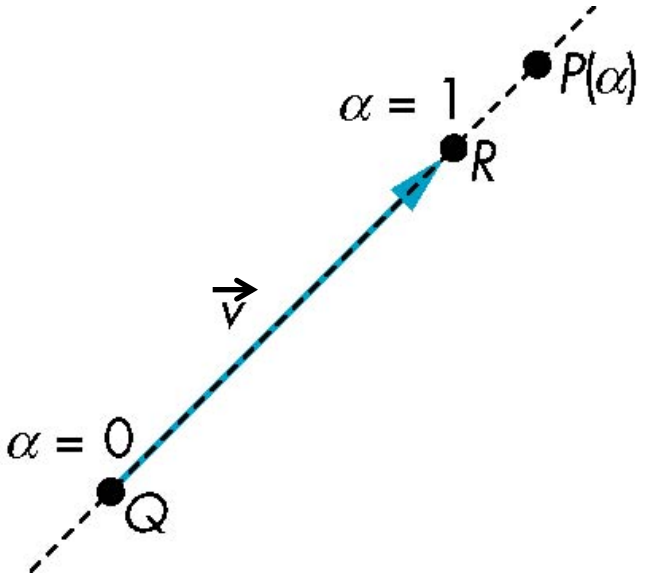
$$y(\alpha) = \alpha y_0 + (1-\alpha)y_1$$

Fundamentos

- Rayos y segmentos de línea:
  - Si $\alpha >= 0$, entonces $P(\alpha)$ es el rayo que parte de P_0 en la dirección de $\vec{\mathbf{V}}$
  - Si se usan dos puntos para definir $\vec{\mathbf{V}}$, entonces:

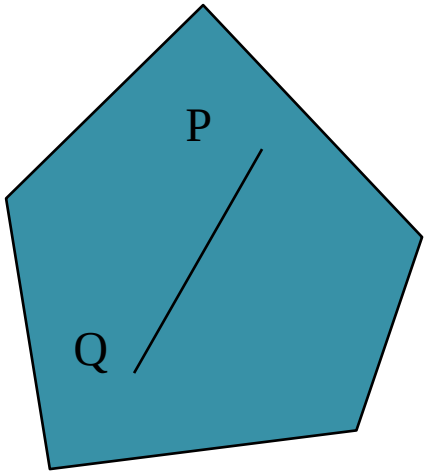
$$P(\alpha) = Q + \alpha (R-Q) = Q + \alpha \vec{\mathbf{V}} = \alpha R + (1-\alpha) Q$$

Para $0 <= \alpha <= 1$ se obtienen todos los puntos del segmento de línea que une R y Q

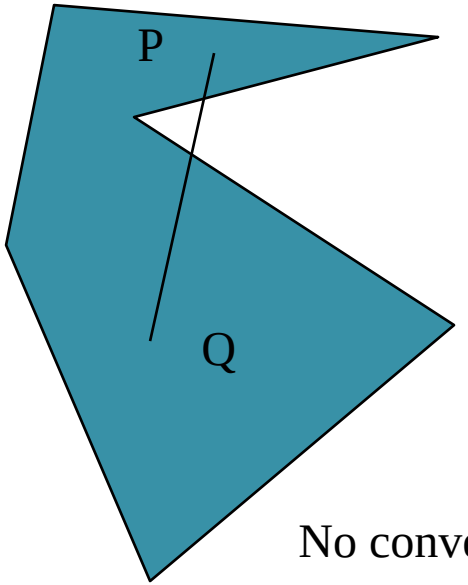


Fundamentos

- Convexidad: Un objeto es convexo si y solo si para cualquier dos puntos dentro del objeto, todos los puntos del segmento de recta entre ellos, también están dentro del objeto



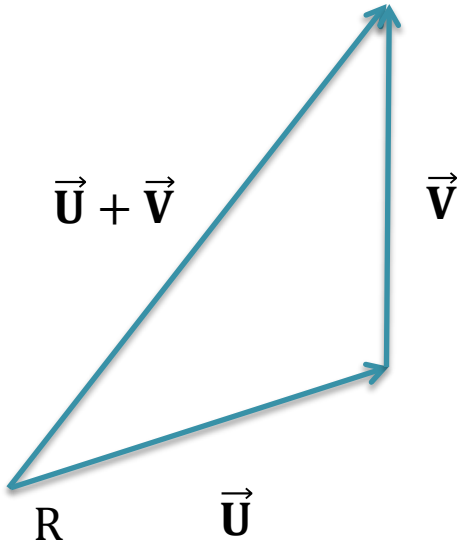
Convexo



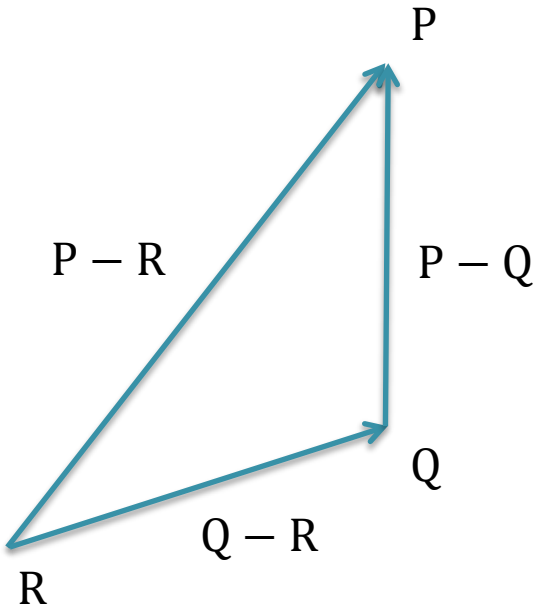
No convexo

Fundamentos

- Un plano se define mediante un punto y dos vectores o mediante tres puntos



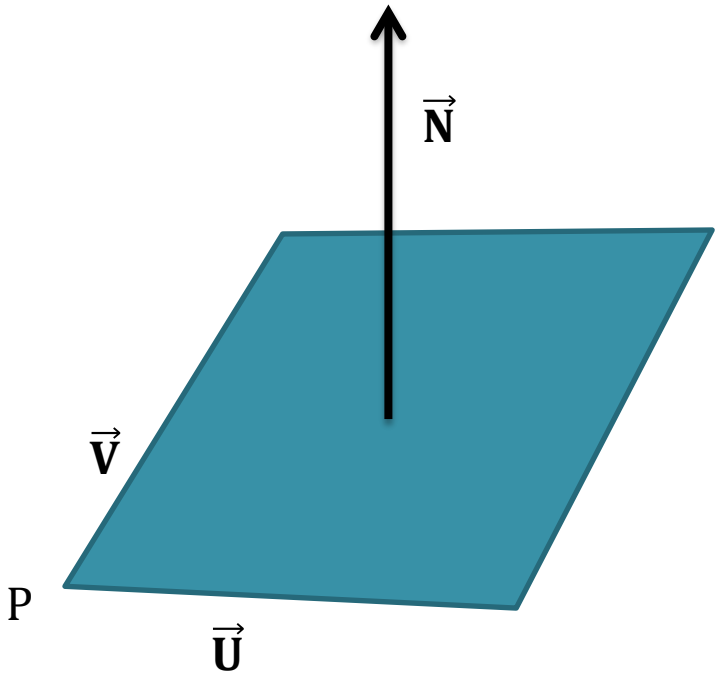
$P(\alpha,\beta)=R+\alpha\,\vec{U}+\beta\,\vec{V}$



$P(\alpha,\beta)=R+\alpha(Q-R)+\beta(P-Q)$

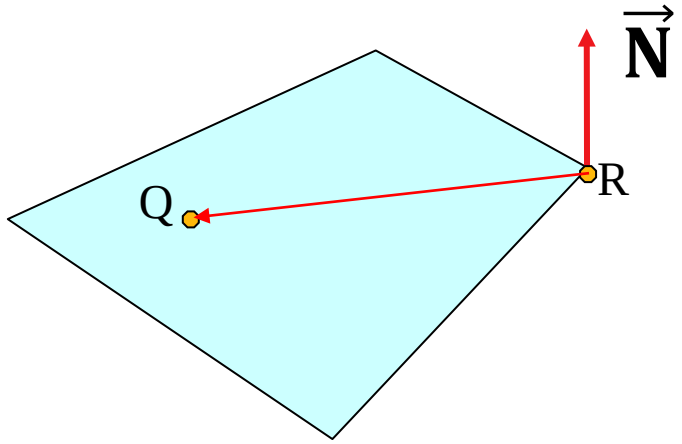
Fundamentos

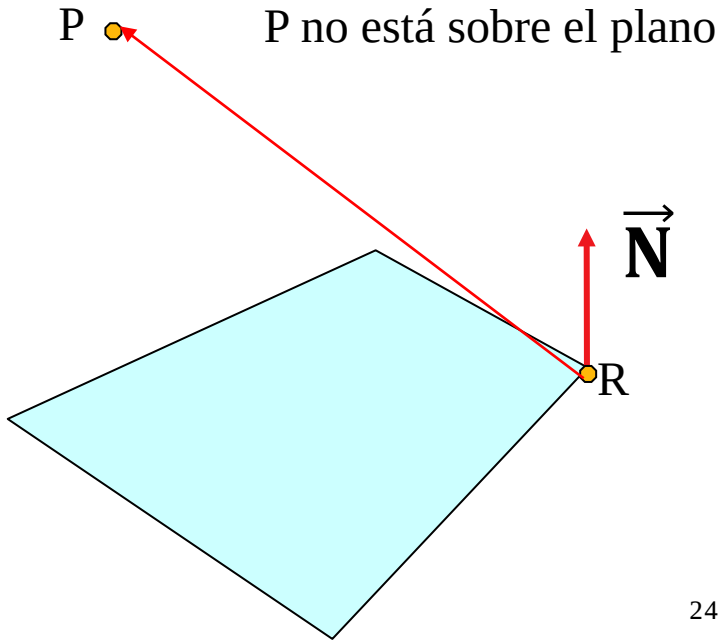
- Cada plano tiene un vector normal a él (perpendicular)
- Este vector se calcula como el producto vectorial de los vectores en la dirección de dos aristas adyacentes: $\vec{\mathbf{N}} = \vec{\mathbf{U}} \times \vec{\mathbf{v}}$



Fundamentos

- Representación implícita del plano:
  - Sea el plano $(R, \vec{N})$ que pasa por el punto R y con normal $\vec{N}$
  - Dado un punto Q sobre el Plano$(R, \vec{N})$ entonces $(Q-R)\bullet\vec{N}=0$
  - La dirección de $\vec{N}$ define la orientación del plano

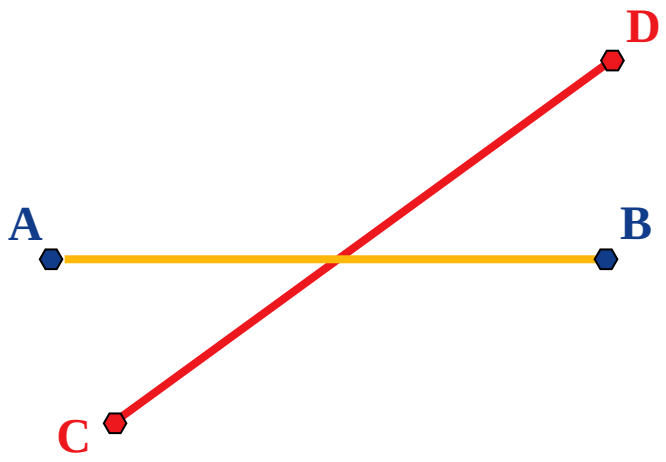


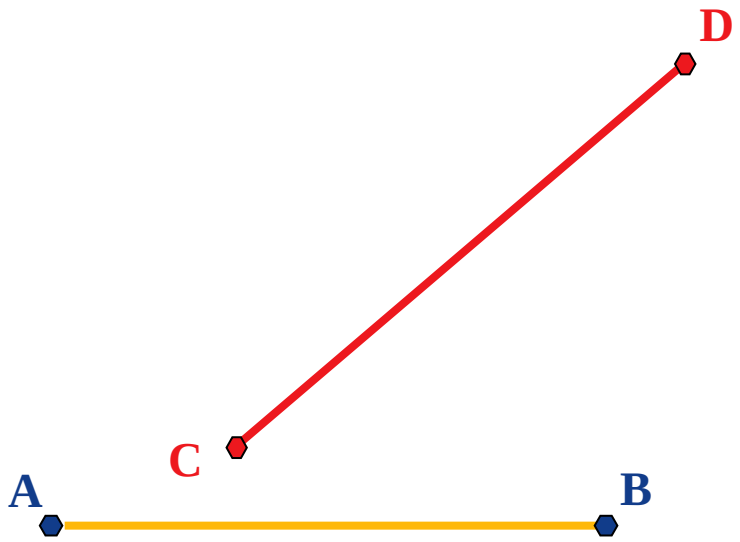


Fundamentos

- Intersección de dos rectas en 2d:

$$0 \geq (\overrightarrow{AB} \times \overrightarrow{AC}) \bullet (\overrightarrow{AB} \times \overrightarrow{AD}) \text{ y } 0 \geq (\overrightarrow{CD} \times \overrightarrow{CA}) \bullet (\overrightarrow{CD} \times \overrightarrow{CB})$$

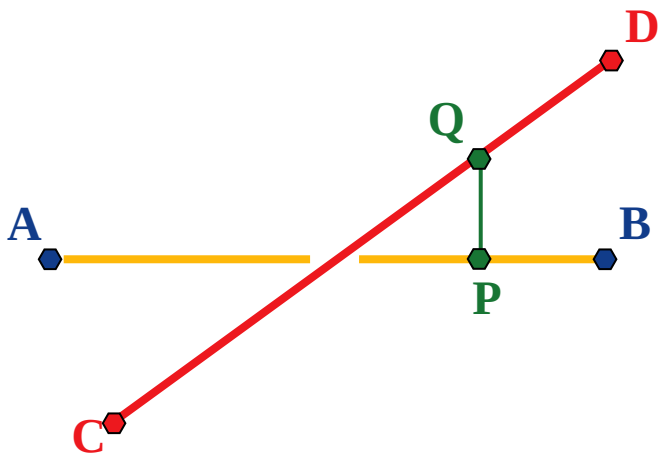


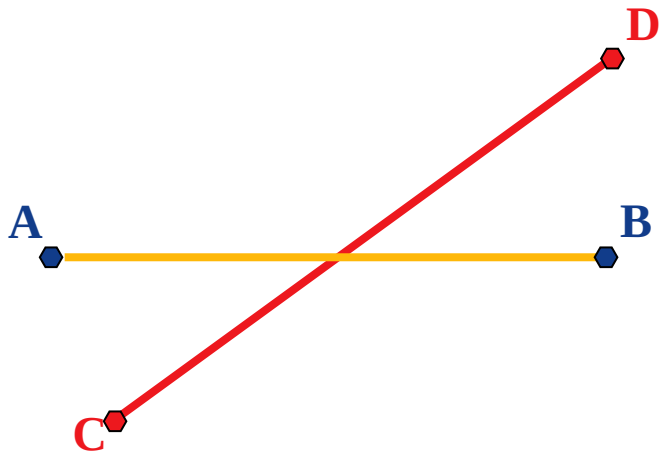


Fundamentos

- Intersección de dos rectas en 3d:
  - Primero deben cumplir que sean coplanares:
    - $\overrightarrow{DA} \bullet (\overrightarrow{DB} \times \overrightarrow{DC}) == 0$
  - Y después se aplica la intersección en 2D

$$0 \geq (\overrightarrow{AB} \times \overrightarrow{AC}) \bullet (\overrightarrow{AB} \times \overrightarrow{AD}) \text{ y } 0 \geq (\overrightarrow{CD} \times \overrightarrow{CA}) \bullet (\overrightarrow{CD} \times \overrightarrow{CB})$$

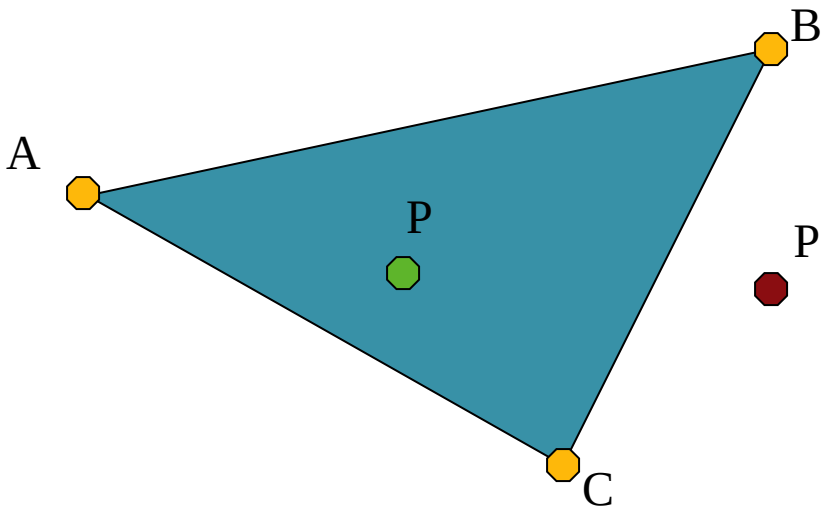




Fundamentos

- ¿Cuando un punto P está en el interior de un triángulo de vértices A,B,C?
  - Cuando A,B,C y P son coplanares y

$$(\overrightarrow{AB}\times\overrightarrow{AP})\cdot(\overrightarrow{BC}\times\overrightarrow{BP})\geq 0 \text{ Y } (\overrightarrow{BC}\times\overrightarrow{BP})\cdot(\overrightarrow{CA}\times\overrightarrow{CP})\geq 0$$



Fundamentos

- La intersección Q de una Recta (P, $\vec{V}$) con un Plano (R, $\vec{N}$) se obtiene:

Reemplazando Q por P+s$\vec{V}$en (Q–R)•$\vec{N}$=0

nos da (P–R+s$\vec{V}$)•$\vec{N}$=0

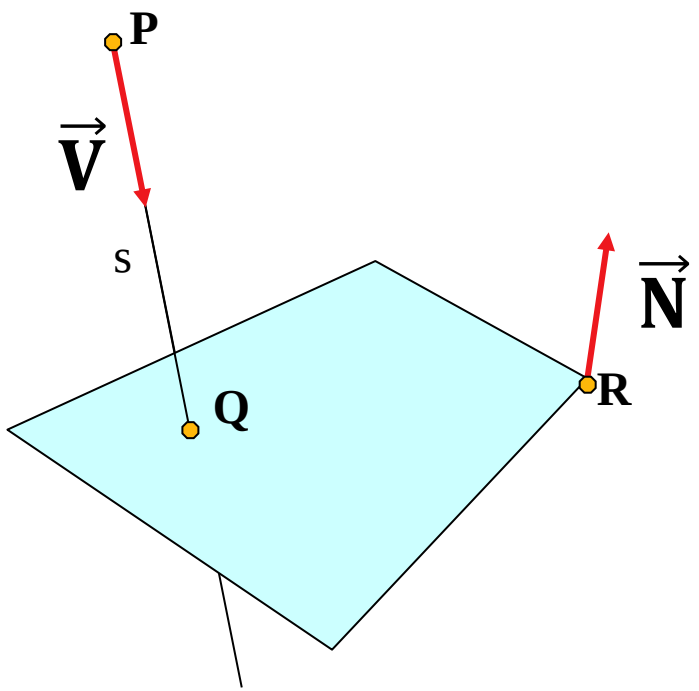
Despejando s

(P-R)•$\vec{N}$+s$\vec{V}$•$\vec{N}$=0 y

s = – (P-R)•$\vec{N}$ /$\vec{V}$•$\vec{N}$

s = (R-P)•$\vec{N}$ /$\vec{V}$•$\vec{N}$

Entonces Q = P + ((R-P)•$\vec{N}$)$\vec{V}$/$\vec{V}$•$\vec{N}$



Fundamentos

- Producto de matrices
  - Solo se pueden multiplicar 2 matrices (**A·B**) si el número de columnas de la primera (**A**) es igual al número de filas de la segunda (**B**)

$$\mathbf{A}_{3x2} = \begin{bmatrix} 0 & -1 \\ 5 & 7 \\ -2 & 8 \end{bmatrix} \qquad \mathbf{B}_{2x2} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\mathbf{A}\cdot\mathbf{B} = \begin{bmatrix} 0\cdot 1+(-1)\cdot 3 & 0\cdot 2+(-1)\cdot 4 \\ 5\cdot 1+7\cdot 3 & 5\cdot 2+7\cdot 4 \\ -2\cdot 1+8\cdot 3 & -2\cdot 2+8\cdot 4 \end{bmatrix} = \begin{bmatrix} -3 & -4 \\ 26 & 38 \\ 22 & 28 \end{bmatrix}$$

  - El producto de matrices no es conmutativo

$$\mathbf{A}\cdot\mathbf{B} \neq \mathbf{B}\cdot\mathbf{A}$$

Fundamentos

- Producto de vector por matriz
  - Es un caso particular del producto de 2 matrices

$$\mathbf{M}_{3x3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad \mathbf{V}_{3x1} = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix}$$

$$\mathbf{M} \cdot \mathbf{V} = \begin{bmatrix} 1 \cdot 0 + 2 \cdot 5 + 3 \cdot (-2) \\ 4 \cdot 0 + 5 \cdot 5 + 6 \cdot (-2) \\ 7 \cdot 0 + 8 \cdot 5 + 9 \cdot (-2) \end{bmatrix} = \begin{bmatrix} 4 \\ 13 \\ 22 \end{bmatrix}$$

Fundamentos

- Traspuesta de una matriz
  - Se obtiene intercambiando filas por columnas

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \mathbf{M}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\left(\mathbf{M}^T\right)^T = \mathbf{M}$$

Transformaciones 2D

- Puntos 2D: como vectores columna
- Transformaciones: matrices cuadradas que premultiplican al vector
- Si considerásemos los puntos como vectores fila, utilizaríamos las matrices traspuestas

$$\mathbf{M} \bullet \vec{\mathbf{V}} \approx \vec{\mathbf{V}}^{\mathsf{T}} \mathbf{M}^{\mathsf{T}}$$

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \end{pmatrix} = \begin{pmatrix} 14 \\ 22 \end{pmatrix}$$

$$\begin{pmatrix} 5 & 3 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 14 & 22 \end{pmatrix}$$

Transformaciones 2D

Traslación

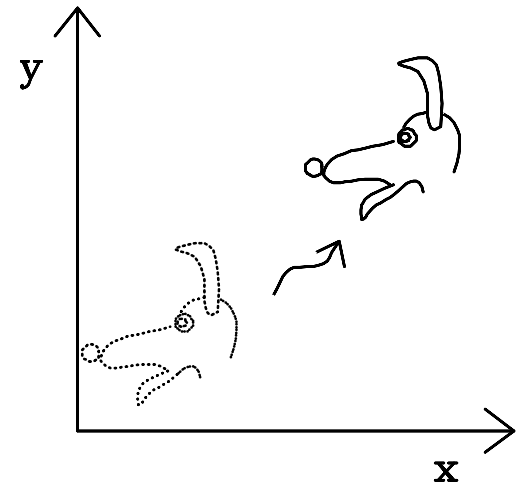
- Desplazar un objeto desde una posición a otra diferente

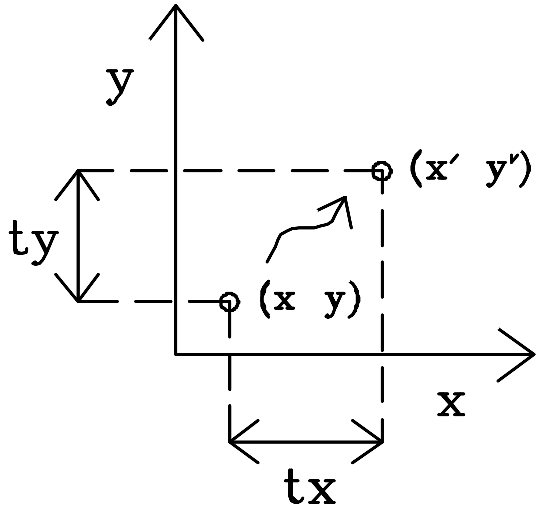
$$x' = x + tx;\ y' = y + ty;$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = P + T$$





- Borrar la primitiva de la posición actual
- Sumar desplazamiento a los puntos de la primitiva
- Redibujar la primitiva en la nueva posición

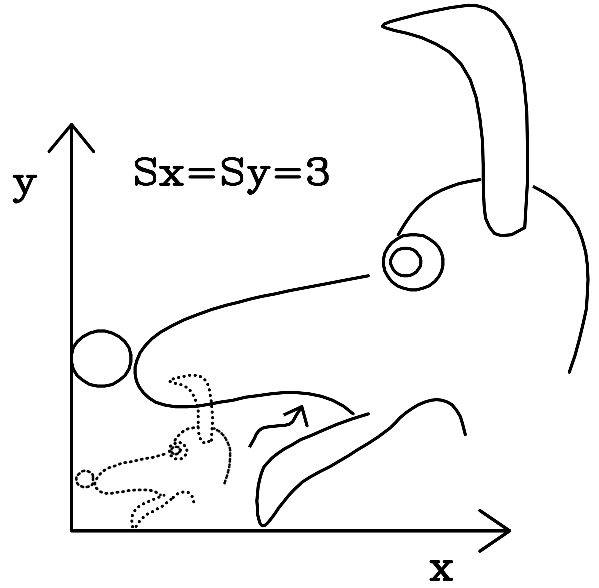
Transformaciones 2D
Escalado

- Modificación del tamaño de un objeto

$$x' = x * sx;\ y' = y * sy;$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$$

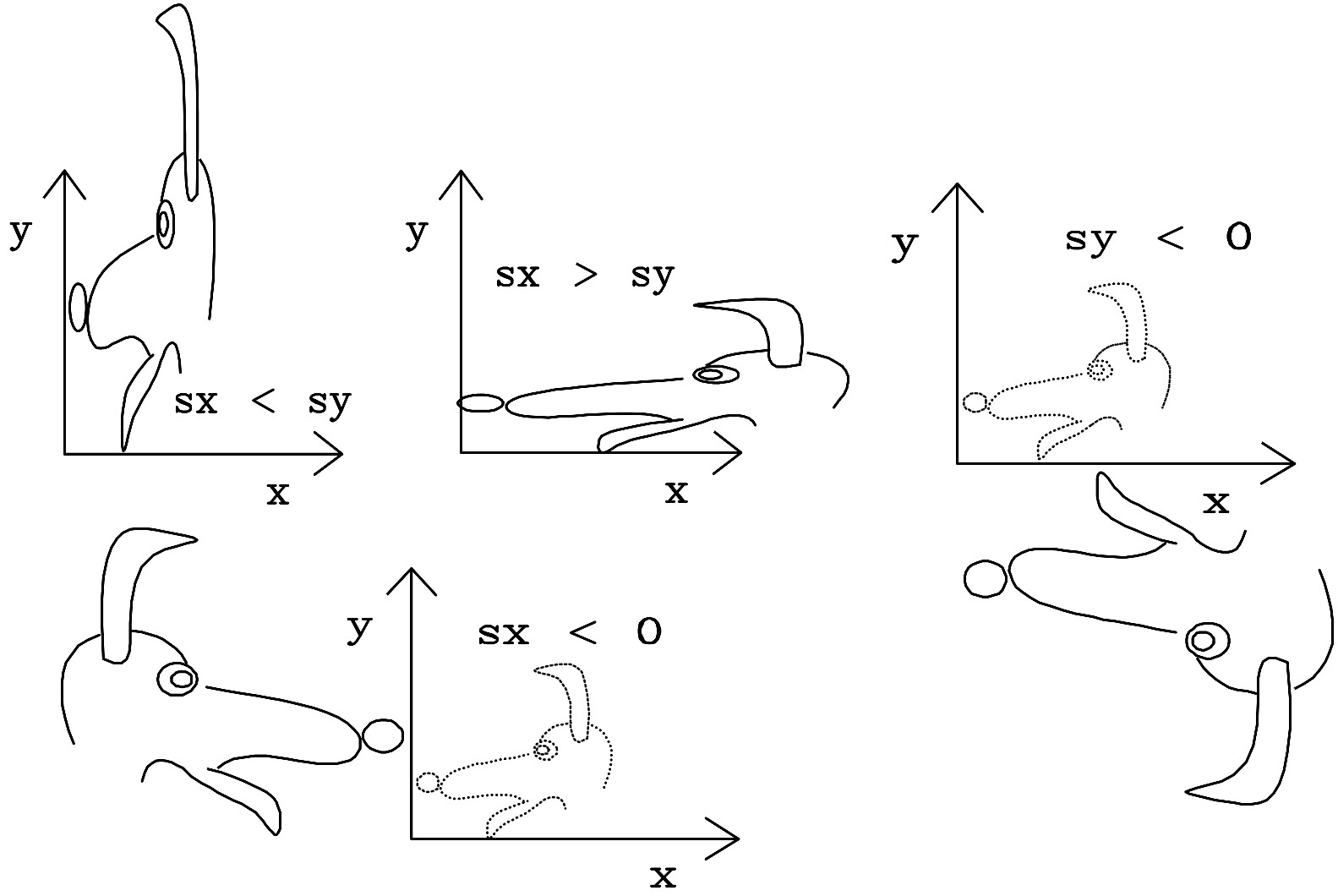


$$P' = \mathbf{S} * P$$

- Si un factor de escala es menor que 1 el objeto se reduce en esa dimensión
- Si un factor de escala es mayor que 1 el objeto aumenta en esa dimensión
- Si un factor de escala es negativo el objeto se invierte en esa dimensión
- Si los factores de escala son diferentes se cambian las proporciones
- El escalado se realiza respecto a un punto fijo (en este caso el origen)

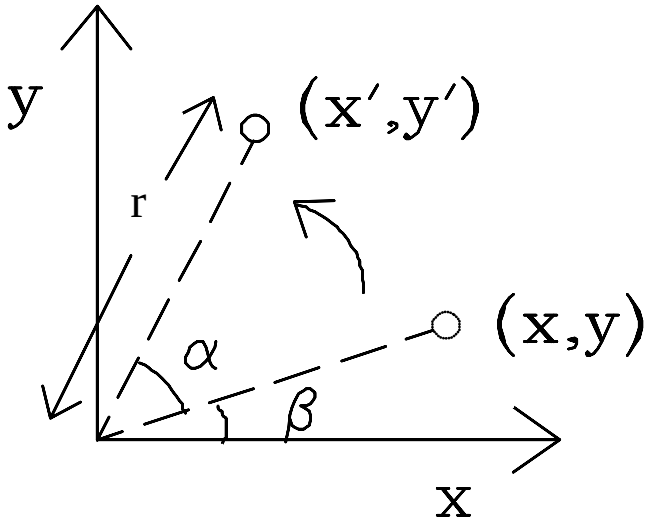
Transformaciones 2D

Escalado



Transformaciones 2D
Rotación

- Giro de un punto respecto de otro (en este caso el origen):
  - Las coordenadas cartesianas de un punto son (x,y)
  - Sus correspondientes coordenadas polares son:
    - x = r*cos(β)
    - y = r*sin(β)
  - Si aplicamos un giro de α grados respecto al origen al punto (x,y):
    - x' = r*cos(β+α)
    - y' = r*sin(β + α)



Transformaciones 2D
Rotación

- Giro de un punto respecto de otro (en este caso el origen)
  - Si a un punto de coordenadas polares (r, β) se le aplica un giro α:

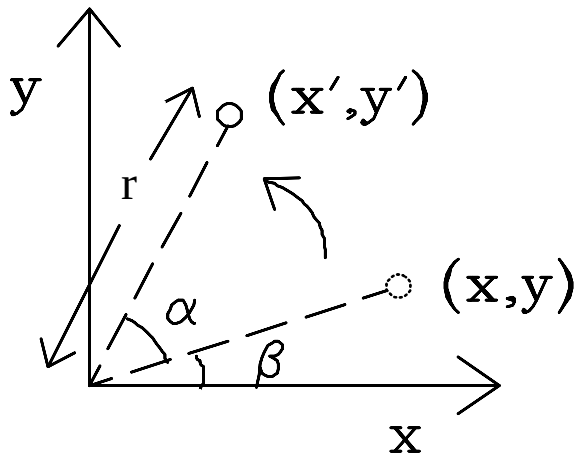
x' = r*cos(β+α) = r*(cos(β)*cos(α) - sin(β)*sen (α)) =

(r*cos(β))*cos(α) - (r*sin(β))*sin (α) = x*cos(α) - y*sin (α)

$$\mathbf{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

y' = r*sin(β + α) = r*(sin(β)*cos(α) + cos(β)*sin (α)) =

(r*sin(β))*cos(α) + (r*cos(β))*sin (α) = y*cos(α) + x*sin (α)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = \mathbf{R} * P$$

Transformaciones 2D

- Representación matricial de las transformaciones:
  - Traslación: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$
  - Escalado: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$
  - Rotación: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}$
- Si queremos aplicar varias transformaciones a un punto:
  - Si la operación es el producto por una matriz cuadrada se pueden acumular en una única matriz:
    - S1*S2*R1*R2*S3=M
    - Pero si incluimos una traslación, ya no es posible

Transformaciones 2D

- Representación matricial de las transformaciones
  - Las transformaciones anteriores se pueden representar como:

$x' = a*x + b*y + c$

$y' = d*x + e*y + f$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}$$

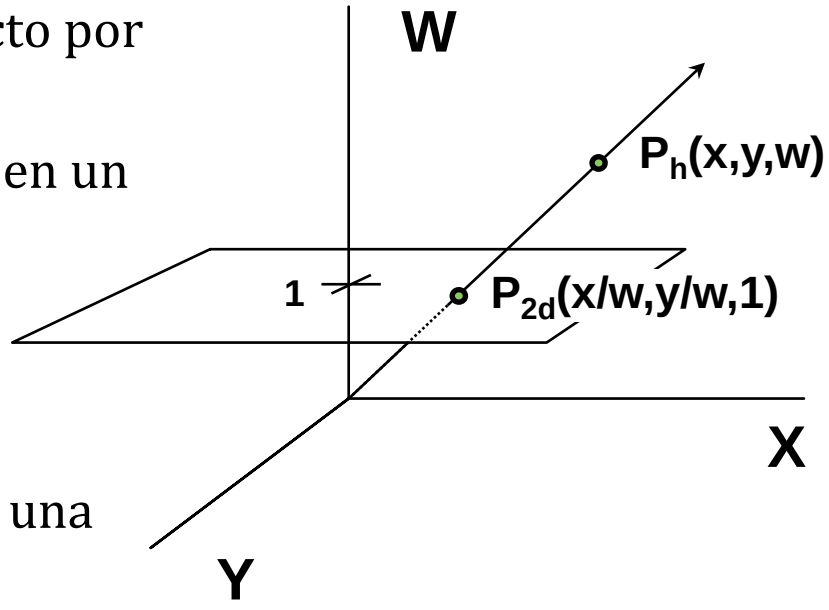
  - Esto se puede representar utilizando matrices
  - Si incluimos todas las constantes en una matriz
  - Es más eficiente manejar matrices cuadradas

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformaciones 2D

- **Coordenadas homogéneas**
  - Conseguimos homogeneizar las tres transformaciones: producto por una matriz cuadrada
  - Para obtener las coordenadas homogéneas se incluye el plano en un espacio 3D cuya tercera dimensión es constante

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



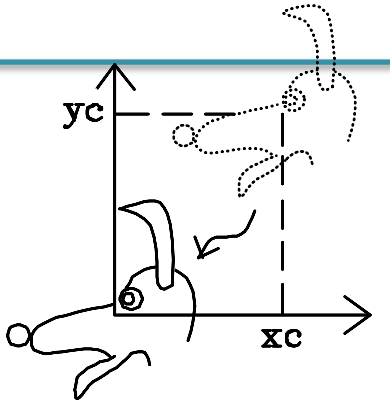
  - De este modo se pueden acumular varias transformaciones en una sola matriz
  - Las nuevas matrices serán:

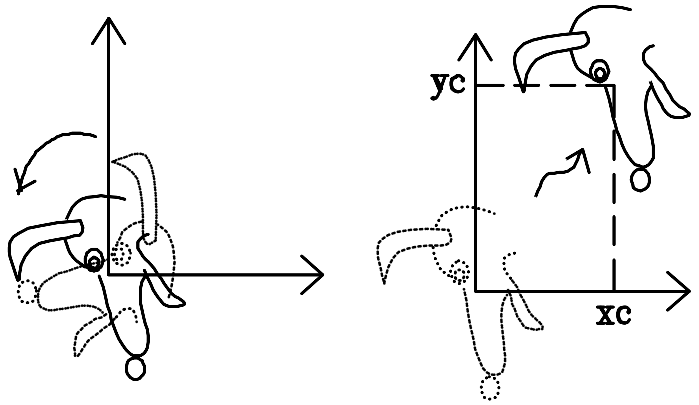
$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} e_x & 0 & 0 \\ 0 & e_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformaciones 2D

- De esta manera podemos combinar varias transformaciones para obtener operaciones más complejas
  - Por ejemplo -> Rotación respecto a un punto cualquiera (x_c , y_c)
    - En tres pasos: Traslación ($-x_c$, $-y_c$), Rotación y Traslación (x_c , y_c)
    - Como las matrices son cuadradas se obtiene una única matriz

      $P3 = \mathbf{T}(x_c , y_c) * \mathbf{R} * \mathbf{T}(-x_c , -y_c) * P$

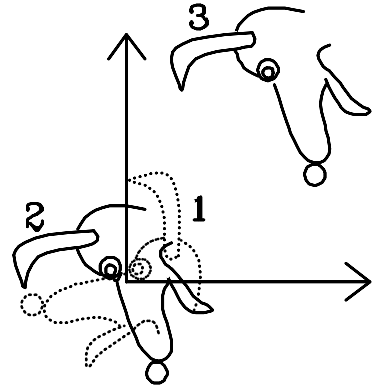




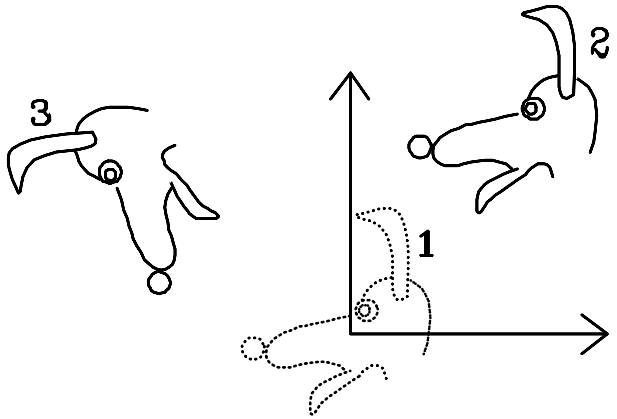
$$\begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & (x_c - \cos(\alpha) \bullet x_c + \sin(\alpha) \bullet y_c) \\ \sin(\alpha) & \cos(\alpha) & (y_c - \sin(\alpha) \bullet x_c - \cos(\alpha) \bullet y_c) \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformaciones 2D

- **Orden de las transformaciones**
  - El producto de matrices no es conmutativo **M1*M2<>M2*M1**
  - La aplicación de transformaciones tampoco lo es
  - Transformaciones que sí son conmutativas
    - Traslación-Traslación
    - Escalado-Escalado
    - Rotación-Rotación
    - Escalado Proporcional-Rotación
  - Transformaciones que no son conmutativas
    - Traslación-Escalado
    - Traslación-Rotación
    - Escalado No Proporcional-Rotación



Traslación después de rotación

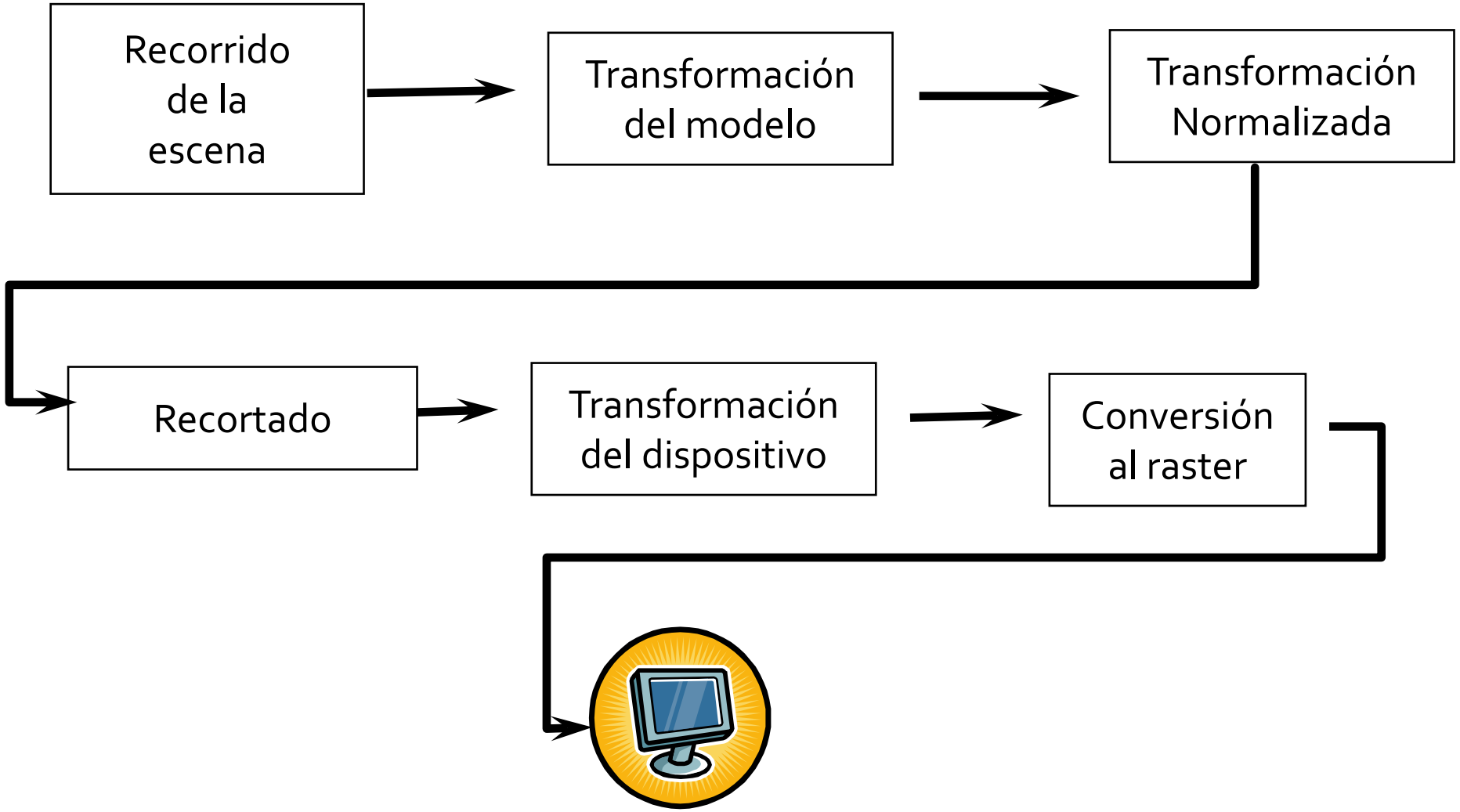


Rotación después de traslación

Transformaciones 2D

- Las operaciones complejas se dividen en otras más sencillas
- ***Traslación***: La traslación no depende de un punto de referencia
- ***Escalado***: El escalado se realiza respecto a un punto de referencia (x,y)
  - Trasladar el objeto (-x,-y)
  - Realizar el escalado del objeto
  - Deshacer la traslación
- ***Rotación***: Las rotaciones se realizan desde un punto llamado centro de rotaciones (CR)
  - Trasladar el objeto (-CR)
  - Realizar la rotación del objeto
  - Deshacer la traslación

Visualización 2D

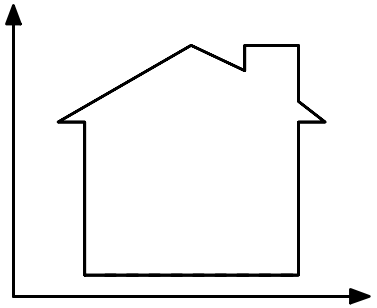


Visualización 2D

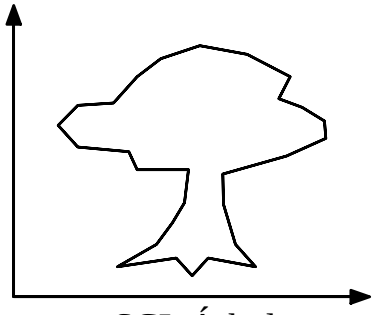
- **Sistema de coordenadas del mundo real (CMR)**
  - Descripción del sistema físico real
  - Las unidades de medida dependen de la representación
  - Se deben transformar en medidas del dispositivo (mapping)
  - La transformación (mapping) se compone de un escalado y de un desplazamiento
- **Sistema de coordenadas del dispositivo (CD)**
  - Depende del dispositivo
    - Tamaño de la ventana, en pixels si es una pantalla
    - Situación del origen de coordenadas
    - Sentido de avance de cada coordenada
- **Sistema de coordenadas del dispositivo normalizado (CDN)**
  - Para realizar el *mapping* de forma normalizada se utiliza un dispositivo ficticio de anchura 1 y altura 1

Visualización 2D

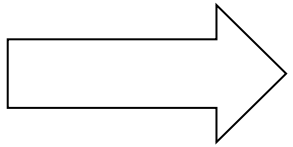
- Componiendo la escena

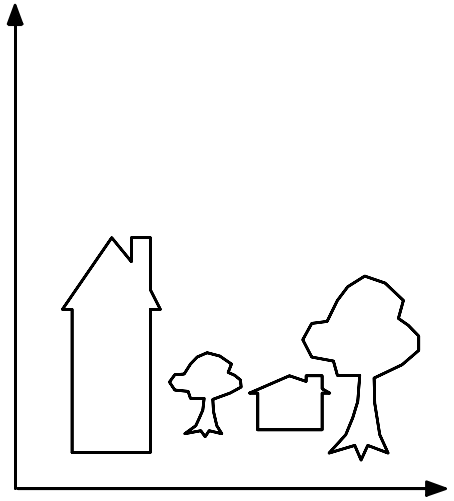


SCL Casa



SCL Árbol

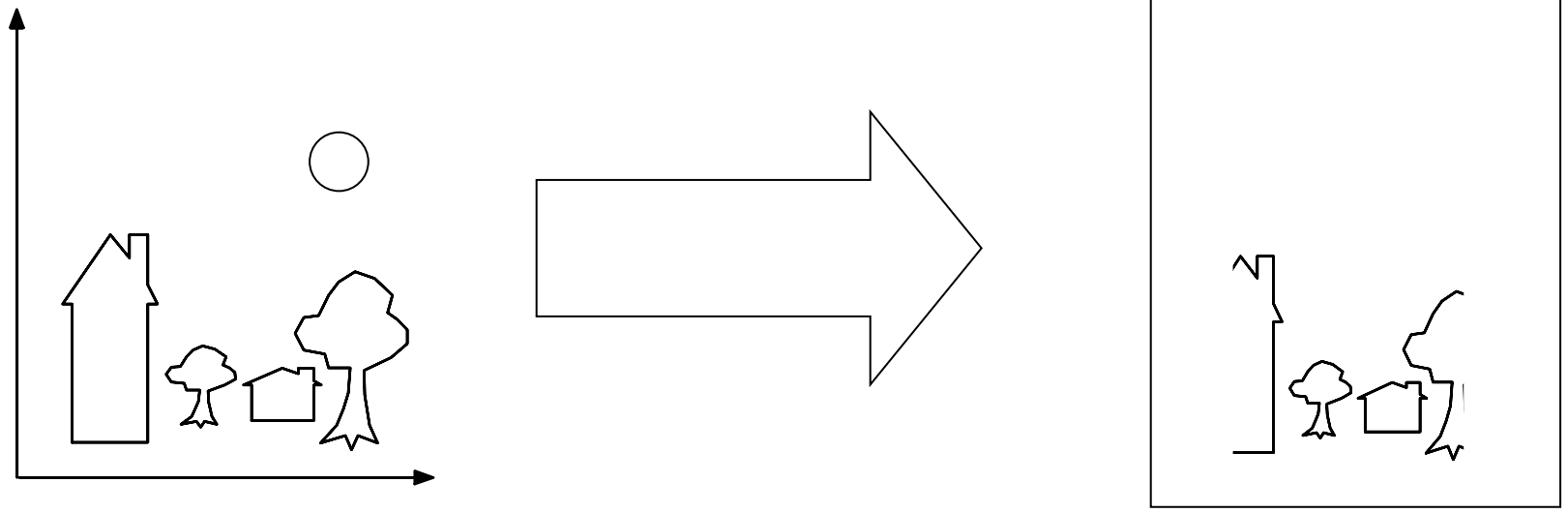




Sistema de Coordenadas del mundo

Visualización 2D

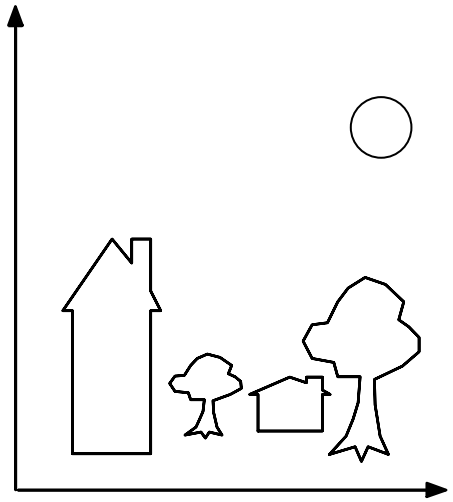
- Problemas de una transformación directa WCS → DCS
  - No se conoce el dispositivo:
    - DPI, origen, …
  - El *mapping* se tiene que reescribir si cambia el dispositivo
  - Es difícil definir el marco en DCS



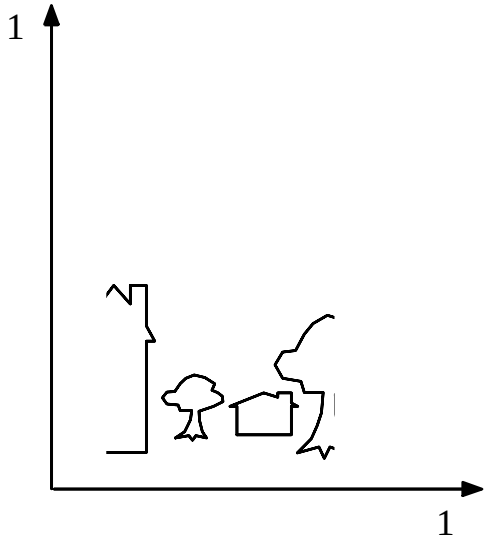
Sistema de coordenadas del mundo

Sistema de coordenadas del dispositivo

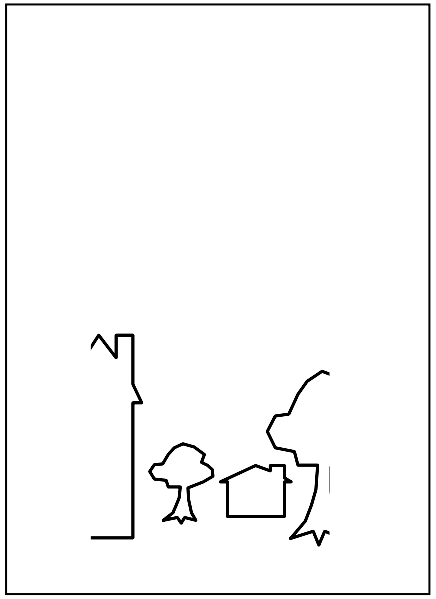
Visualización 2D



Sistema de coordenadas del mundo



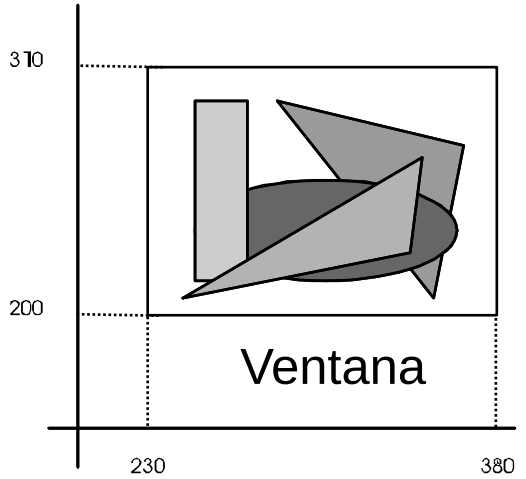
Sistema de coordenadas normalizado



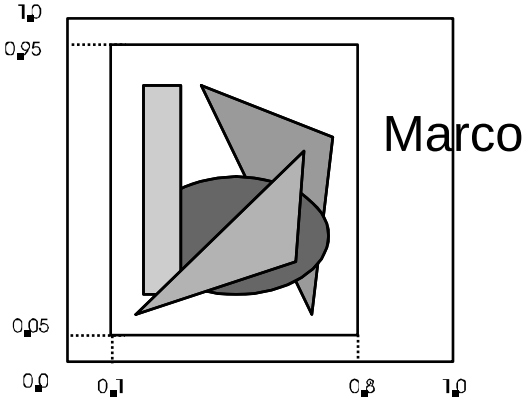
Sistema de coordenadas del dispositivo

Visualización 2D

- Cómo podemos pasar de un sistema de coordenadas a otro de la forma más eficiente y a ser posible de un modo normalizado
- Se define una zona del mundo real: ventana (window)
- Se define una zona del dispositivo: marco (viewport)
  - El tamaño de un marco va de un píxel a toda la pantalla
  - Lo que quede fuera de la ventana queda fuera del marco
- La transformación de coordenadas de la aplicación en coordenadas del dispositivo físico (CD) se realiza en 2 pasos:
  - De CMR a CDN
    - Transformación normalizada
  - De CDN a CD
    - Transformación del dispositivo



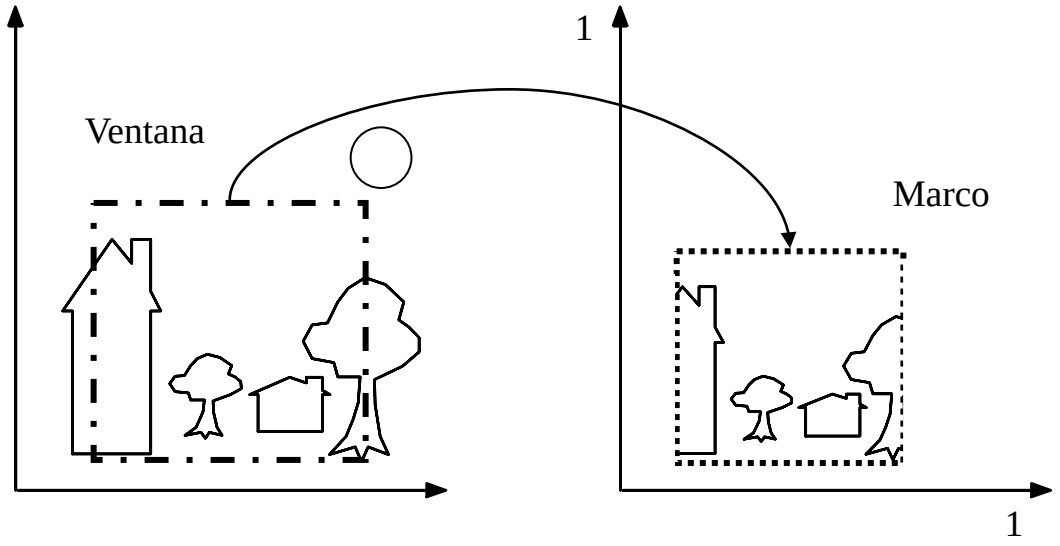
Coordenadas del mundo real



Coordenadas del dispositivo normalizado

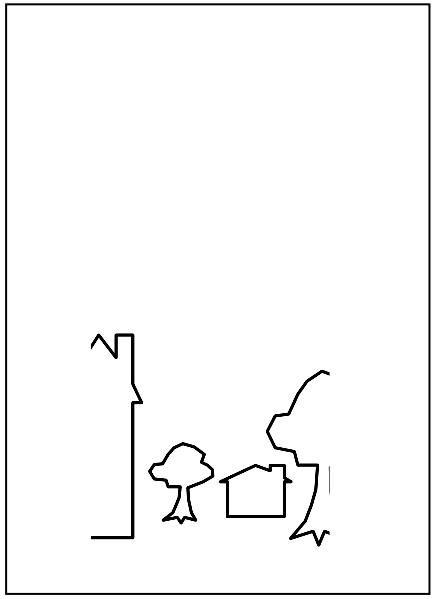
Visualización 2D

- Transformación de coordenadas del mundo a coordenadas normalizadas



Sistema de coordenadas del mundo

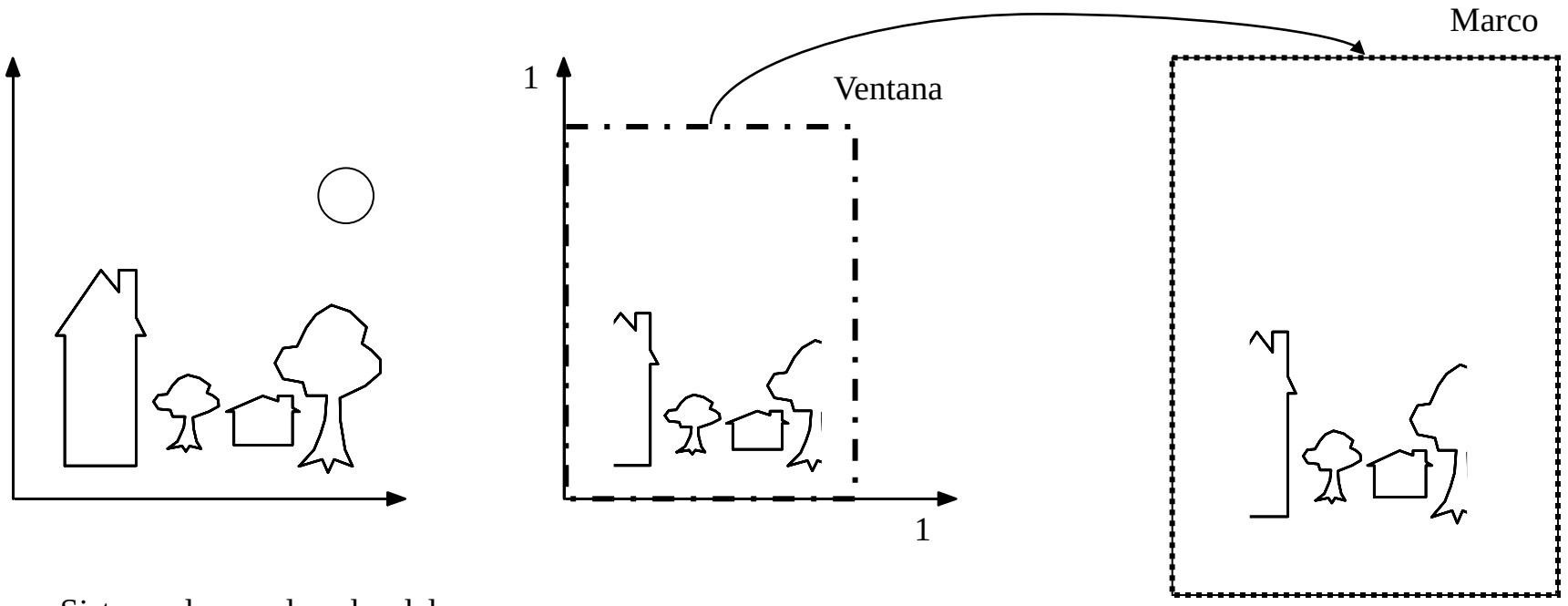
Sistema de coordenadas normalizado



Sistema de coordenadas del dispositivo

Visualización 2D

- Transformación de coordenadas normalizadas a coordenadas del dispositivo



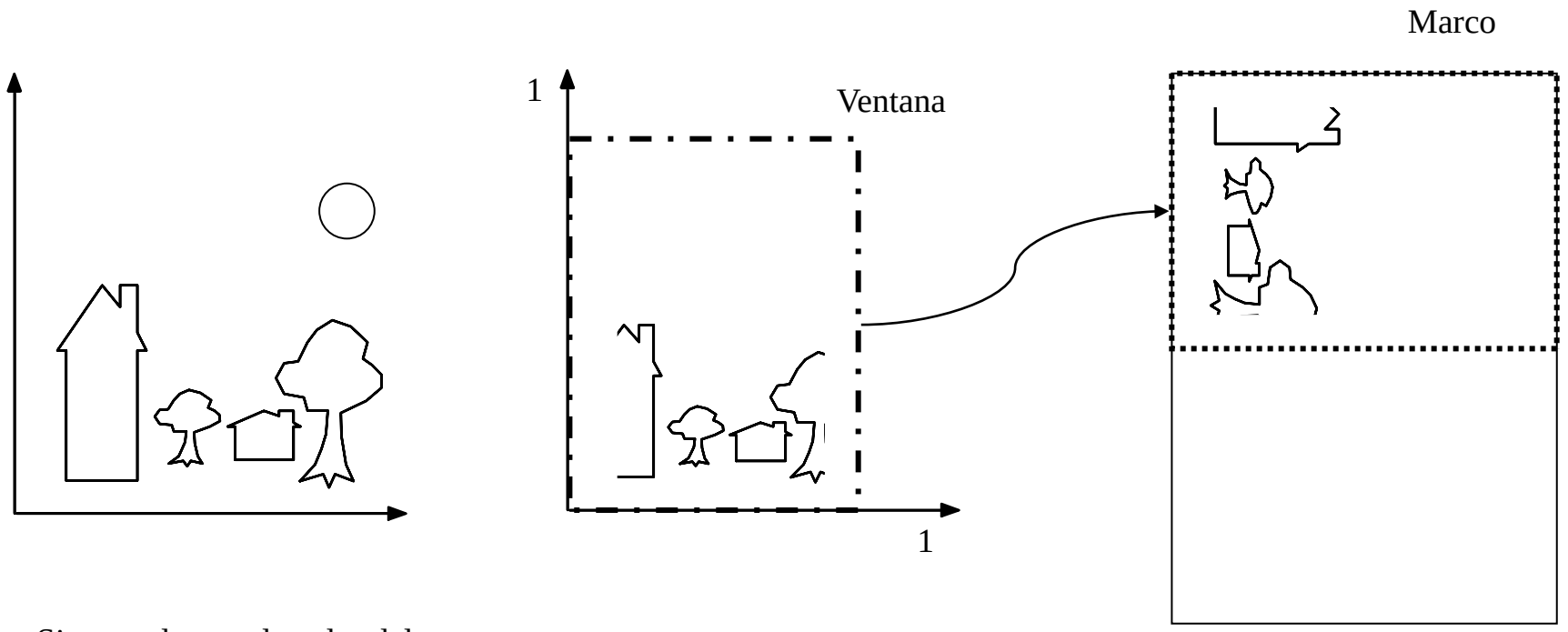
Sistema de coordenadas del mundo

Sistema de coordenadas normalizado

Sistema de coordenadas del dispositivo

Visualización 2D

- Transformación de coordenadas normalizadas a coordenadas del dispositivo



Visualización 2D

- Pasos para transformar de SCM a SCD:

1. Definir el formato final deseado (A4, 1024x768, etc)
   - Calcular el área correspondiente en el dispositivo normalizado
2. Definir la ventana en SCM
   - Qué parte de la escena se dibujará. En coordenadas del mundo
3. Definir el marco en SCDN
   - La posición deseada de la figura en el formato destino
4. Calcular la transformación SCM → SCDN
5. Definir el marco en SCD
   - Permite múltiples páginas por hoja, escalar la imagen, etc.
6. Calcular la transformación SCDN → SCD

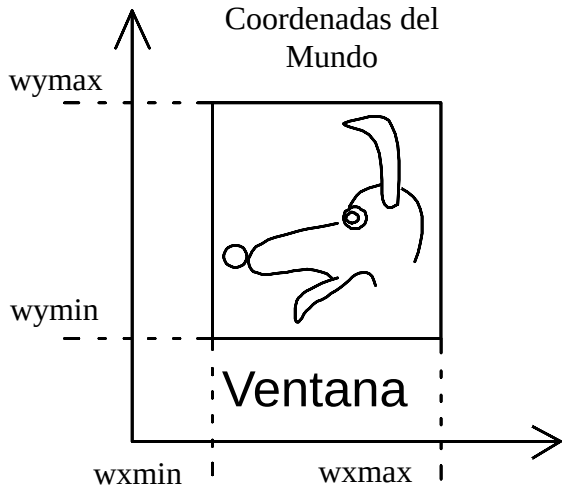
Visualización 2D

- Transformación ventana-marco:
  - Se traslada la esquina inferior izquierda de la ventana al origen
  - Se aplican los factores de escala para que marco y ventana tengan el mismo tamaño
  - Se traslada el origen a la esquina inferior izquierda del marco

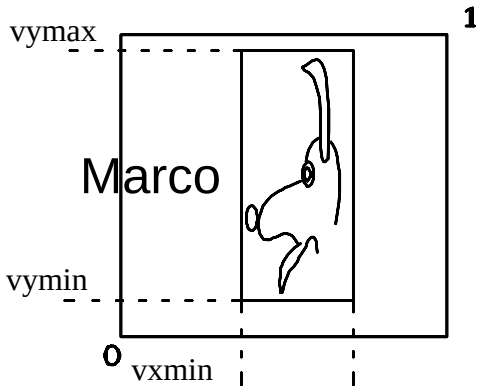
$$s_x = \frac{v_{xmax} - v_{xmin}}{w_{xmax} - w_{xmin}}$$

$$s_y = \frac{v_{ymax} - v_{ymin}}{w_{ymax} - w_{ymin}}$$

$$\mathbf{P'} = \begin{bmatrix} 1 & 0 & v_{xmin} \\ 0 & 1 & v_{ymin} \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & -w_{xmin} \\ 0 & 1 & -w_{ymin} \\ 0 & 0 & 1 \end{bmatrix} \bullet \mathbf{P}$$



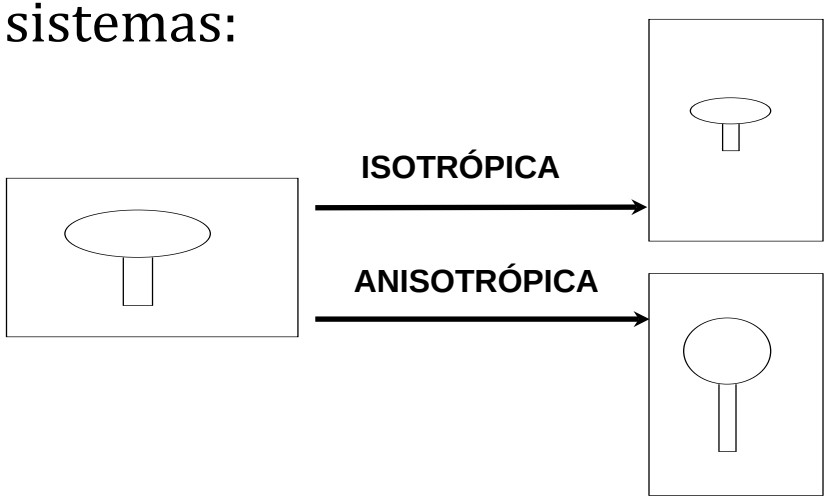
Coordenadas del mundo real



Coordenadas del dispositivo normalizado

Visualización 2D

- La transformación de CDN a CD sería similar a la anterior
- En general las ventanas y los marcos serán rectangulares
- Tipos de transformaciones entre sistemas:



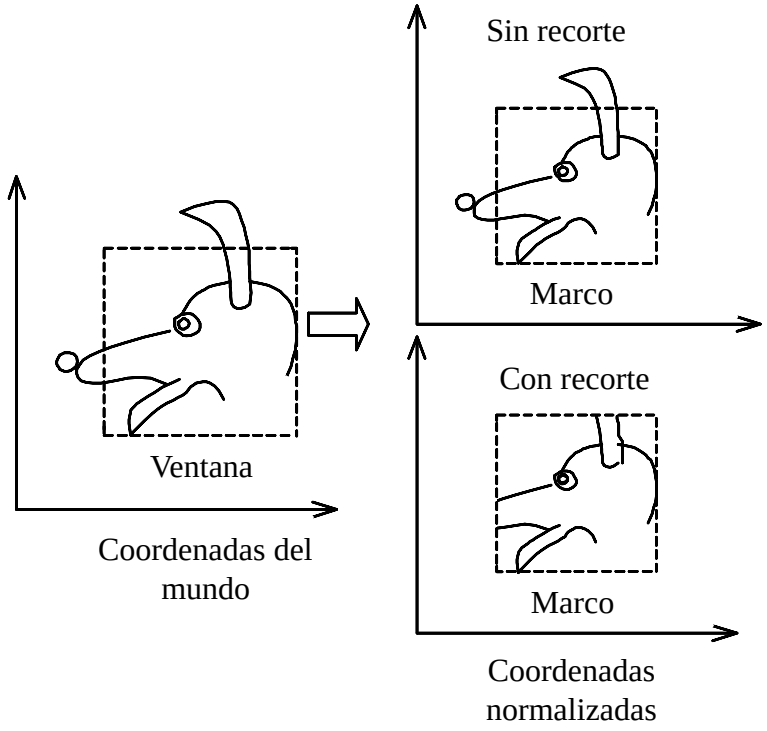
- Isotrópica (conservando proporciones, 1 factor de escala) o anisotrópica (2 factores de escala). Para conseguir la isotrópica, la definición del marco deberá ser proporcional al de la ventana

Visualización 2D

- En este Screencast de Polimedia se explica el cambio de Sistemas de Coordenadas 2D y cómo se aplica en un problema práctico:
- http://hdl.handle.net/10251/83911

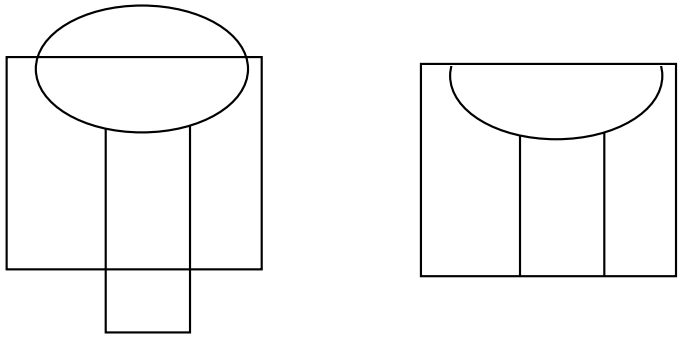
Visualización 2D

- Después de cambiar de CM a CDN, parte de la escena puede permanecer fuera del marco: utilizamos el recortado para quitar esa parte de la escena (no es visible)



Recortado

- Calcula las partes de la escena que son visibles a través de la ventana del mundo real



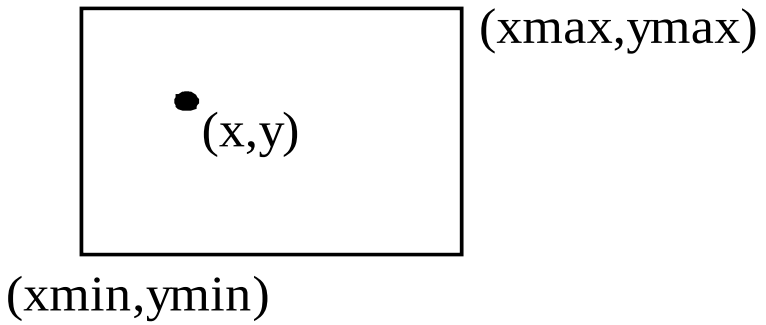
- Se puede hacer de dos formas:
  - Analíticamente: contra la ventana del mundo real
  - Durante la conversión al raster
- Generalmente las ventanas de recortado son rectangulares y con los lados paralelos a los ejes del dispositivo

Recortado

Recortado de puntos

- Es el método de recortado más simple
- Un punto es visible si cumple las siguientes condiciones:

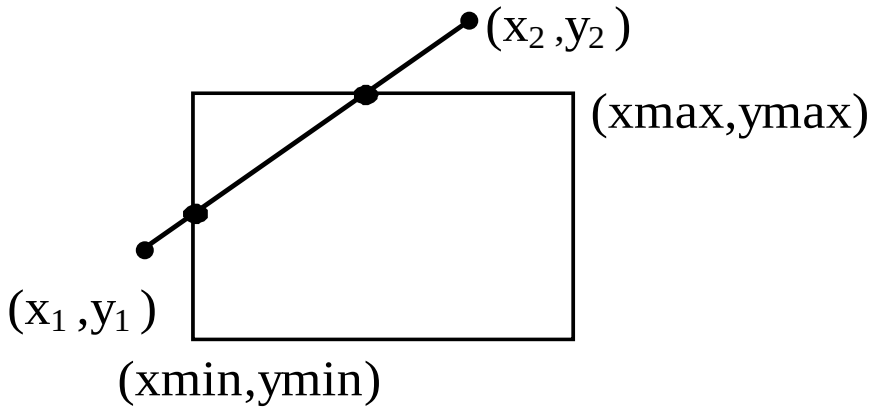
(Xmin <= X <= Xmax) y (Ymin <= Y <= Ymax)



Recortado

Recortado de líneas (contra ventanas rectangulares)

- Para recortar una línea sólo se consideran sus dos extremos

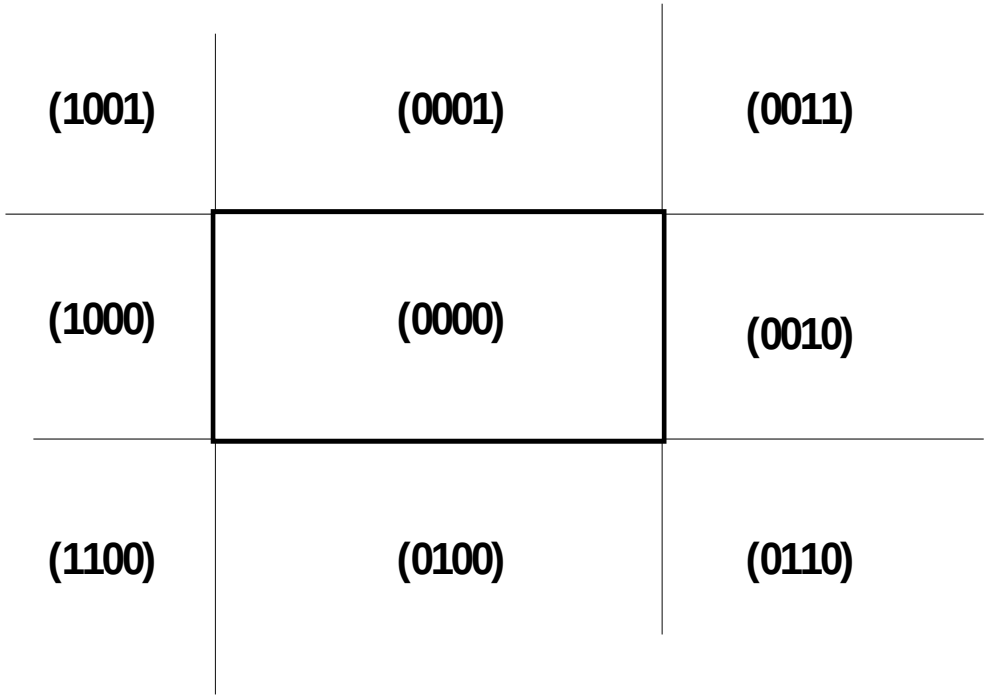


- Tres posibles situaciones:
  - Test trivial de aceptación: si los dos extremos de la línea están dentro de la ventana, la línea también lo está
  - Si sólo un extremo está dentro de la ventana, la línea intersecta el rectángulo: calcular la intersección
  - Ningún extremo está dentro de la ventana: es necesario hacer más cálculos

Recortado

Recortado de líneas (contra ventanas rectangulares)

- Algoritmo de recortado de líneas de Cohen-Sutherland
  - Realiza tests triviales iniciales para evitar calcular intersecciones innecesarias
  - El espacio 2D se divide en nueve zonas, cada una con un código de cuatro bits
  - Cada extremo de una línea se clasifica en una de las nueve zonas



Recortado

Recortado de líneas (contra ventanas rectangulares)

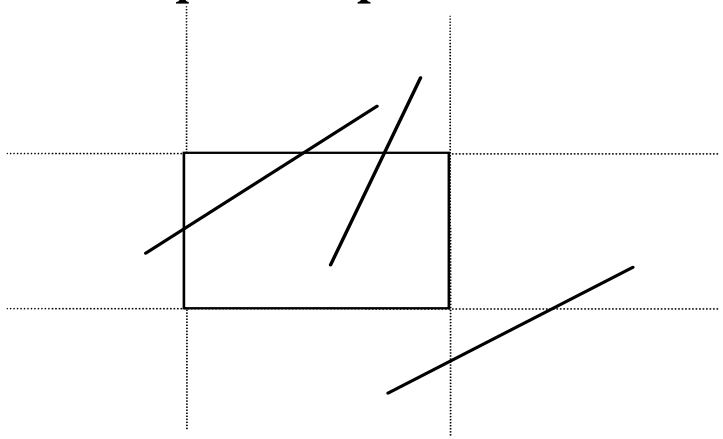
Bit Nº	Comentario	Condición
0	Extremo por encima de la ventana	Y > Ymax
1	Extremo a la derecha de la ventana	X > Xmax
2	Extremo por debajo de la ventana	Y < Ymin
3	Extremo a la izquierda de la ventana	X < Xmin

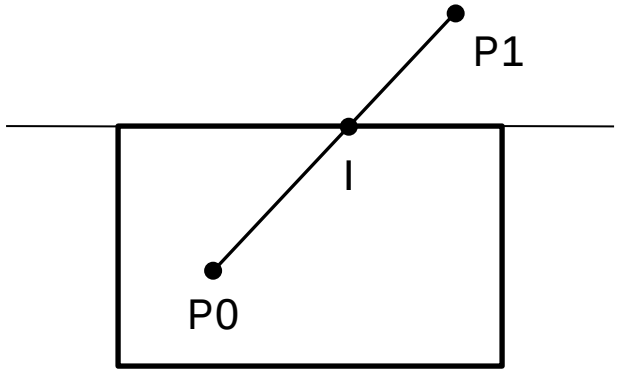
- Se asigna un código de 4 bits a los extremos de la recta (P0,P1):
  - Test trivial de aceptación: CODIGO(P0) OR CODIGO(P1) = 0000
  - Test trivial de rechazo: CODIGO(P0) AND CODIGO(P1) <> 0000 (ambos extremos están en el mismo semiplano)
- Ventajas de los tests triviales:
  - Fáciles de implementar en hardware
  - Se pueden utilizar con cualquier algoritmo de recortado
  - Los bits a 1 indican las fronteras de la ventana que pueden intersectar con la línea (caso de fallar el rechazo trivial)
  - Se pueden aceptar o rechazar rectas sin calcular las intersecciones

Recortado

Recortado de líneas (contra ventanas rectangulares)

- Ejemplos de líneas que no pasan los test triviales



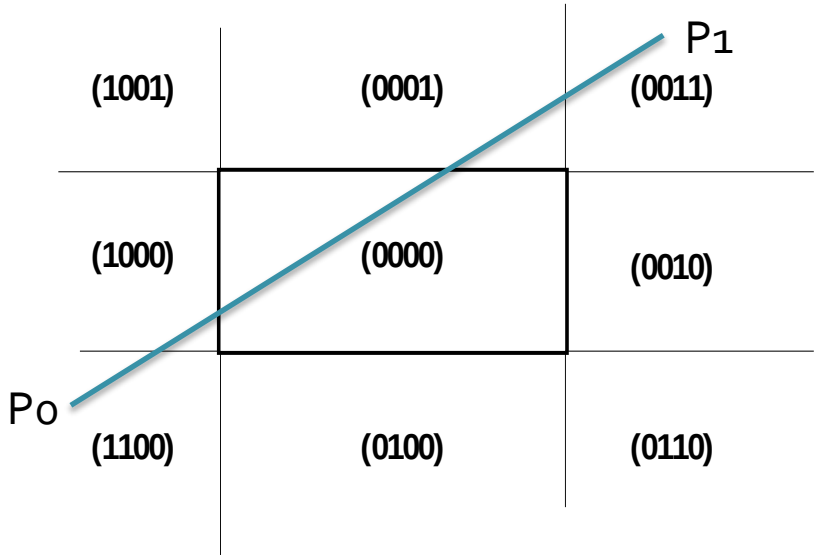


- Si la línea no es aceptada ni rechazada (trivialmente):
  - Elegir un extremo fuera de la ventana
  - Calcular la intersección I de la línea con el límite de la primera región a la que pertenece el extremo elegido (el primer bit a 1 del código del punto)
  - Se elimina el tramo que va desde el extremo elegido a la intersección
  - Si el otro tramo es aceptado trivialmente, el algoritmo termina
  - Si no es aceptado, se calcula la siguiente intersección
- En el peor caso: se calculan las cuatro intersecciones

Recortado

Recortado de líneas (contra ventanas rectangulares)

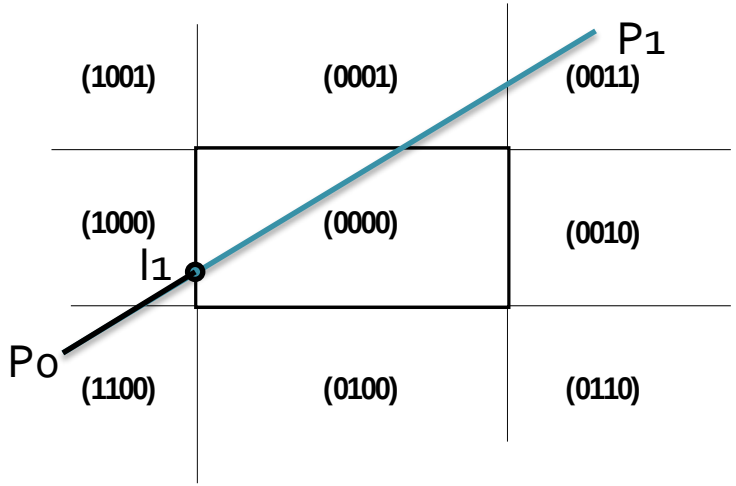
- Ejemplo algoritmo Cohen-Sutherland
  - C(P0)=1100 C(P1)=0011
  - No funcionan test triviales
  - Elijo P0
  - Calcula la Intersección con la región 1000



Recortado

Recortado de líneas (contra ventanas rectangulares)

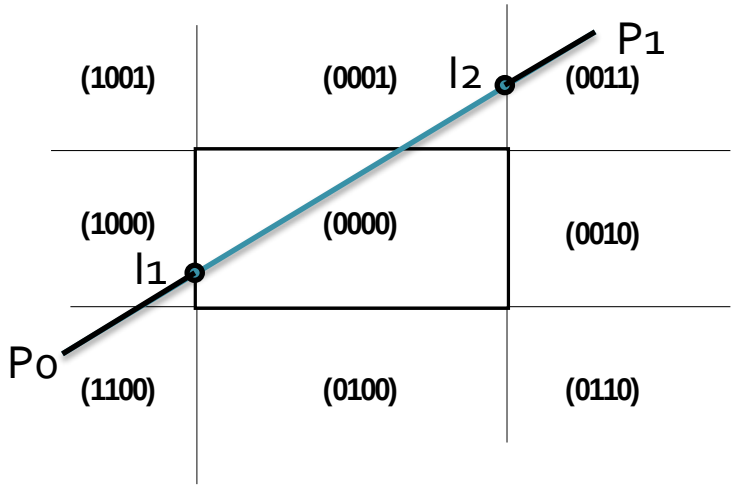
- Se elimina P0-I1
- C(I1)=0000 C(P1)=0011
- I1 ya está en la ventana
- Elijo P1
- Se calcula la intersecciónconcon la región 0010



Recortado

Recortado de líneas (contra ventanas rectangulares)

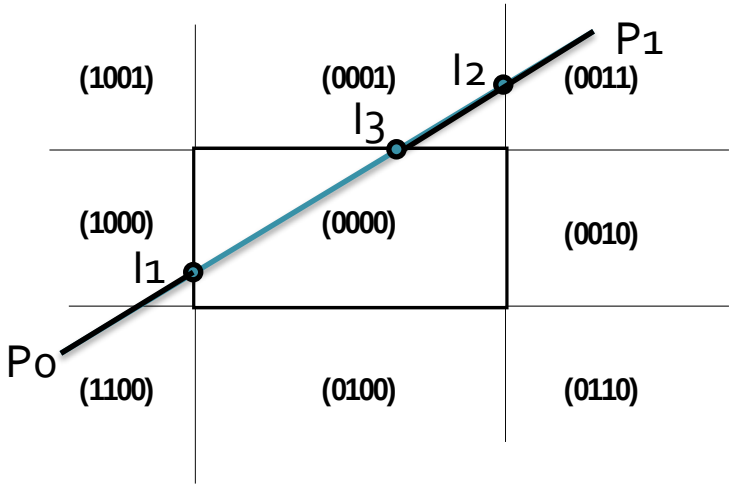
- Se elimina P1-I2
- C(I1)=0000 C(I2)=0001
- Elijo I2
- Se calcula la intersección con la región 0001



Recortado

Recortado de líneas (contra ventanas rectangulares)

- Se elimina I2-I3
- C(I1)=0000 C(I3)=0000
- Se acepta trivialmente



Recortado

Recortado de líneas (contra ventanas rectangulares)

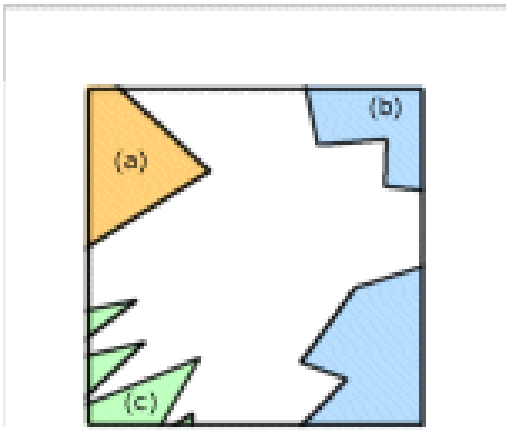
- En este screencast se explica el algoritmo de Cohen-Sutherland para recortado de rectas:
  - Recortado de rectas
  - Algoritmo de Cohen-Sutherland
  - Test triviales
  - Ejercicio

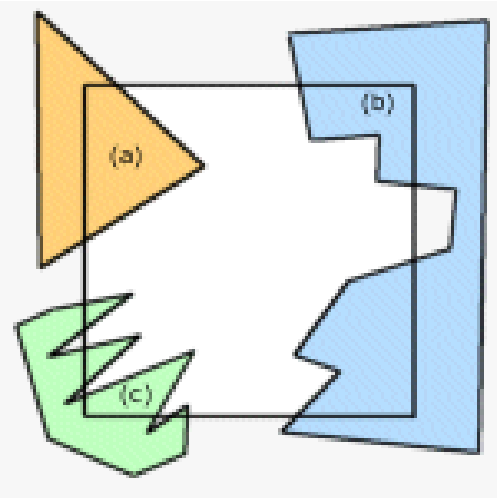
- http://hdl.handle.net/10251/105132

Recortado

Recortado de polígonos

- Dificultad: multiplicidad de casos posibles
- Cada arista del polígono se ha de comprobar contra cada arista de la ventana de recortado
- No es correcto aplicar algoritmos de recortado de aristas





Recortado

Recortado de polígonos

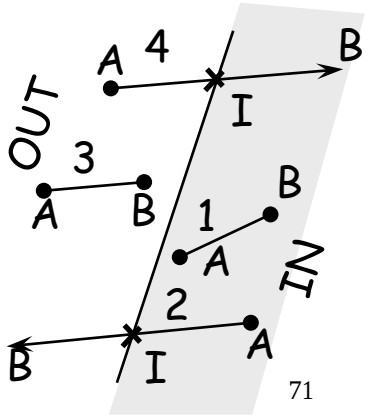
- Algoritmo de Sutherland-Hodgman:
  - Volumen de recorte necesariamente convexo
  - Algoritmo re-entrante para cada arista del área de recorte
    - Se realiza el recortado contra cada una de las aristas de la ventana de recorte
    - Se recorre cada arista del polígono a recortar
    - Se genera una nueva lista de vértices en cada iteración
    - Cuando el algoritmo se ha aplicado a todas las aristas de la ventana el resultado es el polígono recortado
  - Extensión inmediata a 3D

Recortado

Recortado de polígonos

- Cada borde de la ventana (convexa) define dos semiplanos:
  - uno interior (in) y otro exterior (out)
- Entrada: lista de vértices consecutivos del polígono original
- Salida: lista de vértices del polígono recortado
- Algoritmo:
  - Repetir para todas las aristas de la ventana
    - Repetir para todas las aristas del polígono:
    - I: intersección de la arista del polígono con la arista de la ventana

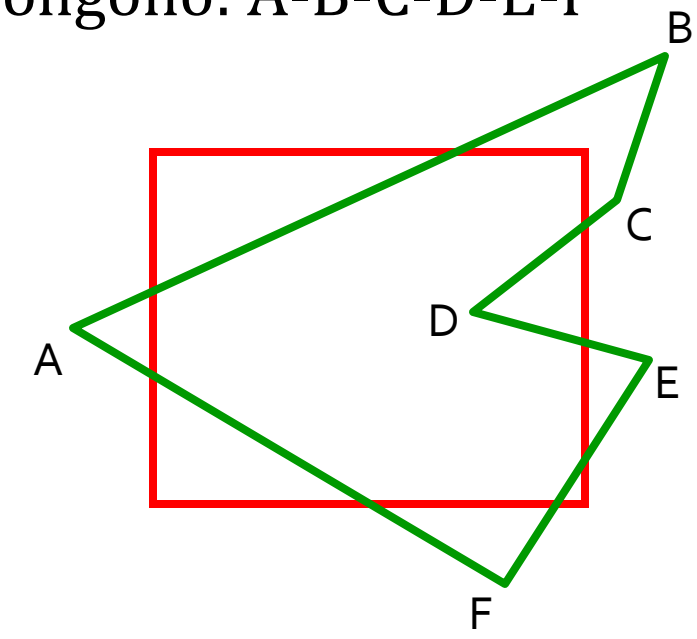
1. Si A in y B in , introducir B en la lista
2. Si A in y B out, introducir I en la lista
3. Si A out y B out, no introducir nada en la lista
4. Si A out y B in , introducir I y B en la lista



Recortado

Recortado de polígonos

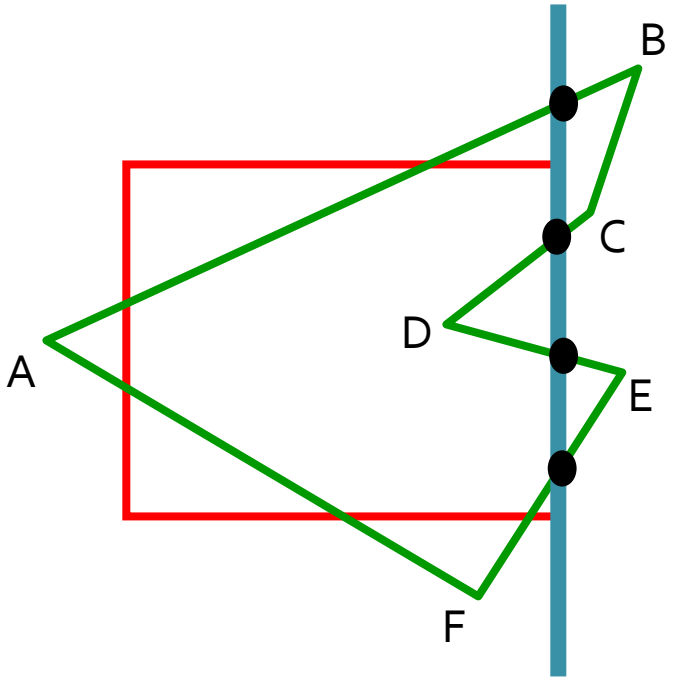
- Idea básica:
  - Se considera cada arista (plano en 3D) de la ventana (volumen en 3D) por separado
  - Se recorta el polígono contra la arista
  - Lista de vértices del polígono: A-B-C-D-E-F

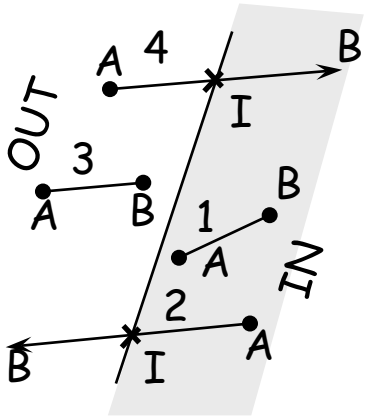


Recortado

Recortado de polígonos

- Recorte arista de la derecha:
  - Polígono inicial: A-B-C-D-E-F
  - A-B CASO 2, se incluye la intersección I1 (I1)
  - B-C CASO 3, no se incluye nada
  - C-D CASO 4, se incluye I2,D (I1-I2-D)
  - D-E CASO 2, se incluye la intersección I3 (I1-I2-D-I3)
  - E-F CASO 4, se incluye I4,F (I1-I2-D-I3-I4-F)
  - F-A CASO 1, se incluye A (I1-I2-D-I3-I4-F-A)
- Polígono de salida: (I1-I2-D-I3-I4-F-A)

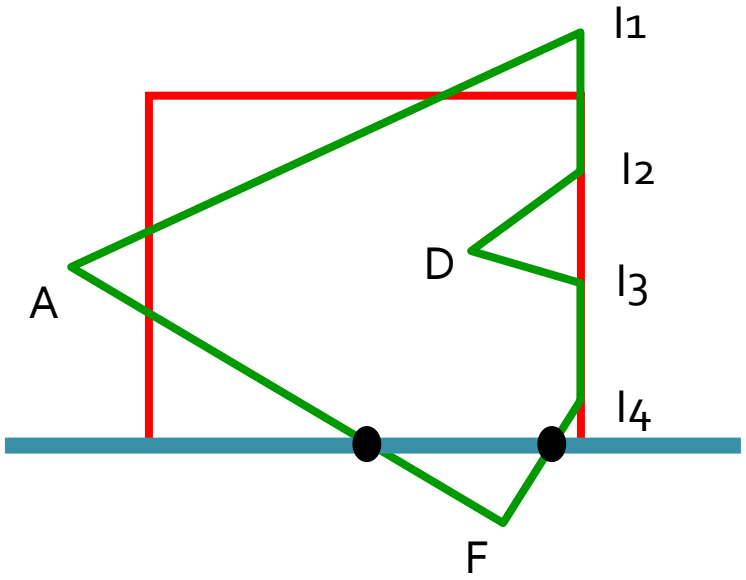


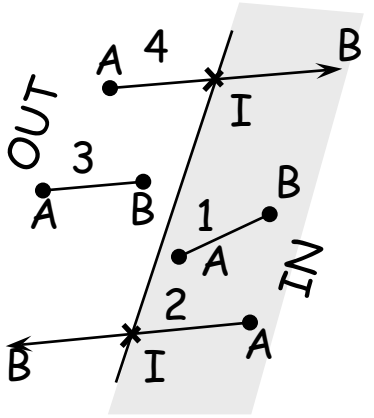


Recortado

Recortado de polígonos

- Recorte arista de abajo:
  - Polígono inicial: I1-I2-D-I3-I4-F-A
  - I1-I2 CASO 1, se incluye I2 (I2)
  - I2-D CASO 1, se incluye D (I2-D)
  - D-I3 CASO 1, se incluye I3 (I2-D-I3)
  - I3-I4 CASO 1, se incluye I4 (I2-D-I3-I4)
  - I4-F CASO 2, se incluye I5 (I2-D-I3-I4-I5)
  - F-A CASO 4, se incluye I6,A (I2-D-I3-I4-I5-I6-A)
  - A-I1 CASO 1, se incluye I1 (I2-D-I3-I4-I5-I6-A-I1)
- Polígono de salida: (I2-D-I3-I4-I5-I6-A-I1)

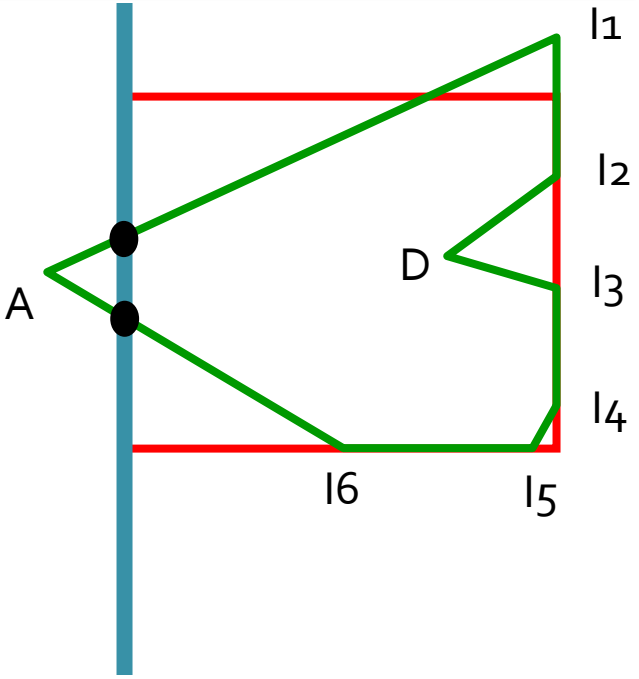


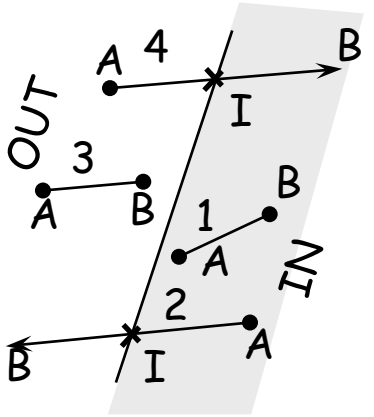


Recortado

Recortado de polígonos

- Recorte arista de la derecho:
  - Polígono inicial: I2-D-I3-I4-I5-I6-A-I1
  - I2-D CASO 1, se incluye D (D)
  - D-I3 CASO 1, se incluye I3 (D-I3)
  - I3-I4 CASO 1, se incluye I4 (D-I3-I4)
  - I4-I5 CASO 1, se incluye I5 (D-I3-I4-I5)
  - I5-I6 CASO 1, se incluye I6 (D-I3-I4-I5-I6)
  - I6-A CASO 2, se incluye I7 (D-I3-I4-I5-I6-I7)
  - A-I1 CASO 4, se incluye I8,I1 (D-I3-I4-I5-I6-I7-I8-I1)
  - I1-I2 CASO 1, se incluye I2 (D-I3-I4-I5-I6-I7-I8-I1-I2)
- Polígono de salida: (D-I3-I4-I5-I6-I7-I8-I1-I2)

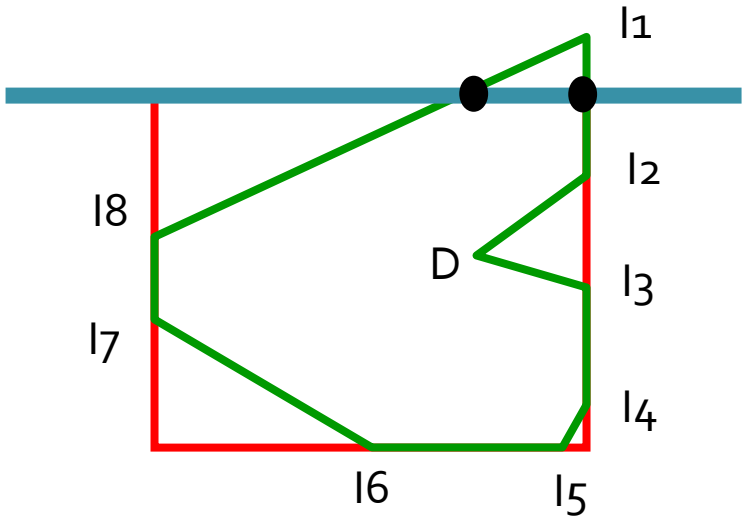


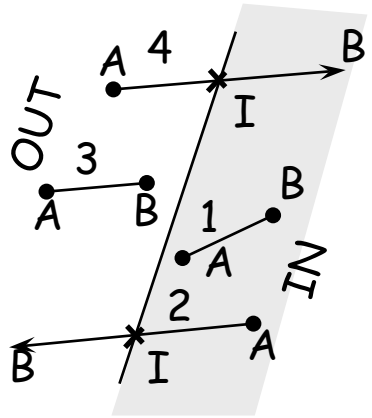


Recortado

Recortado de polígonos

- Recorte arista de arriba:
  - Polígono inicial: D-I3-I4-I5-I6-I7-I8-I1-I2
  - D-I3 CASO 1, se incluye I3 (I3)
  - I3-I4 CASO 1, se incluye I4 (I3-I4)
  - I4-I5 CASO 1, se incluye I5 (I3-I4-I5)
  - I5-I6 CASO 1, se incluye I6 (I3-I4-I5-I6)
  - I6-I7 CASO 1, se incluye I7 (I3-I4-I5-I6-I7)
  - I7-I8 CASO 1, se incluye I8 (I3-I4-I5-I6-I7-I8)
  - I8-I1 CASO 2, se incluye I9 (I3-I4-I5-I6-I7-I8-I9)
  - I1-I2 CASO 4, se incluye I10,I2 (I3-I4-I5-I6-I7-I8-I9-I10-I2)
  - I2-D CASO 1, se incluye D (I3-I4-I5-I6-I7-I8-I9-I10-I2-D)
- Polígono de salida: (I3-I4-I5-I6-I7-I8-I9-I10-I2-D)

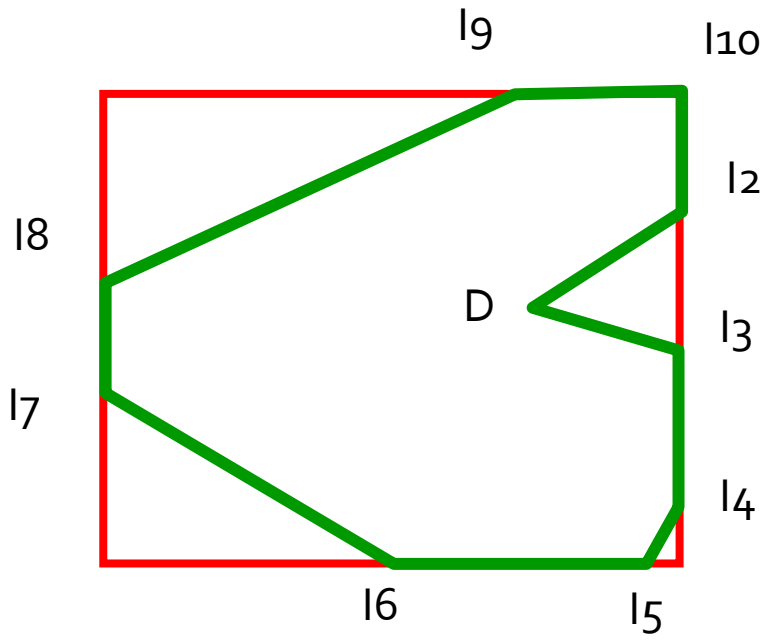




Recortado

Recortado de polígonos

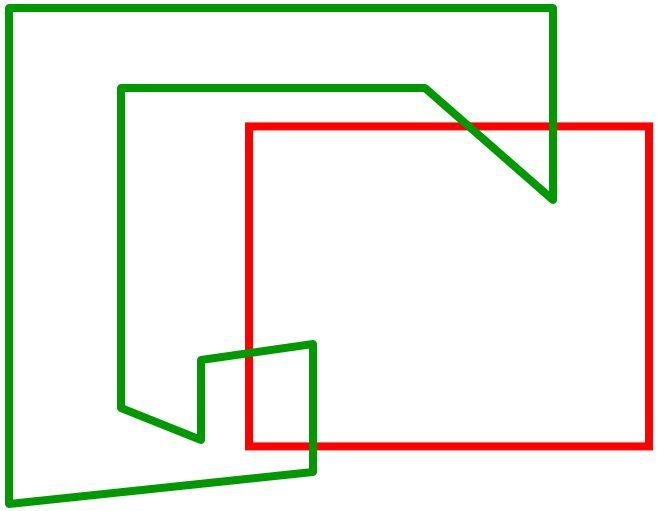
- RESULTADO FINAL:
- Polígono de salida: (I3-I4-I5-I6-I7-I8-I9-I10-I2-D)



Recortado

Recortado de polígonos

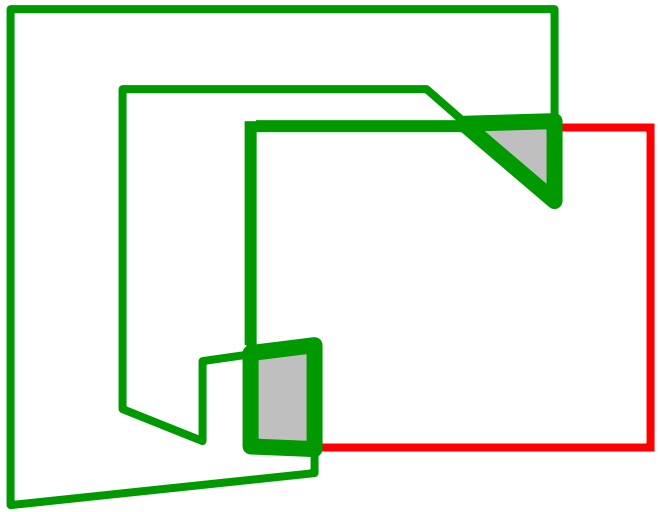
- Recortado complejo: ¿cual sería el resultado de aplicar el algoritmo para recortar este polígono?



Recortado

Recortado de polígonos

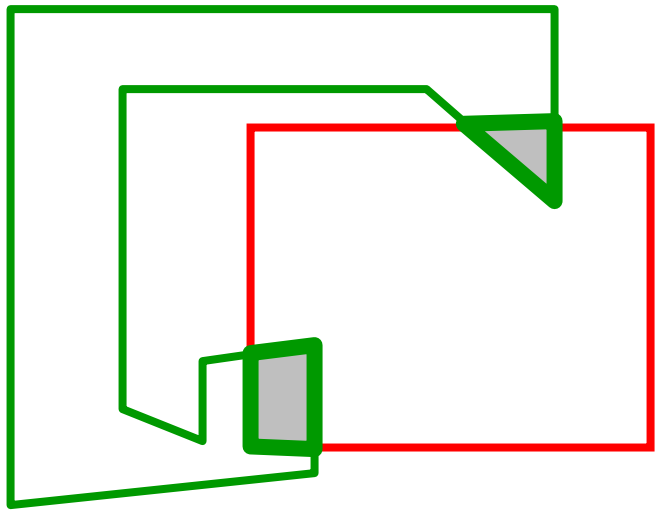
- Resultado: Aparecen aristas fantasmas porque el resultado correcto son dos polígonos y el algoritmo solo devuelve uno.



Recortado

Recortado de polígonos

- Resultado correcto: Es necesario generar más de una lista de polígonos como resultado (Weiler-Atherton)



Recortado

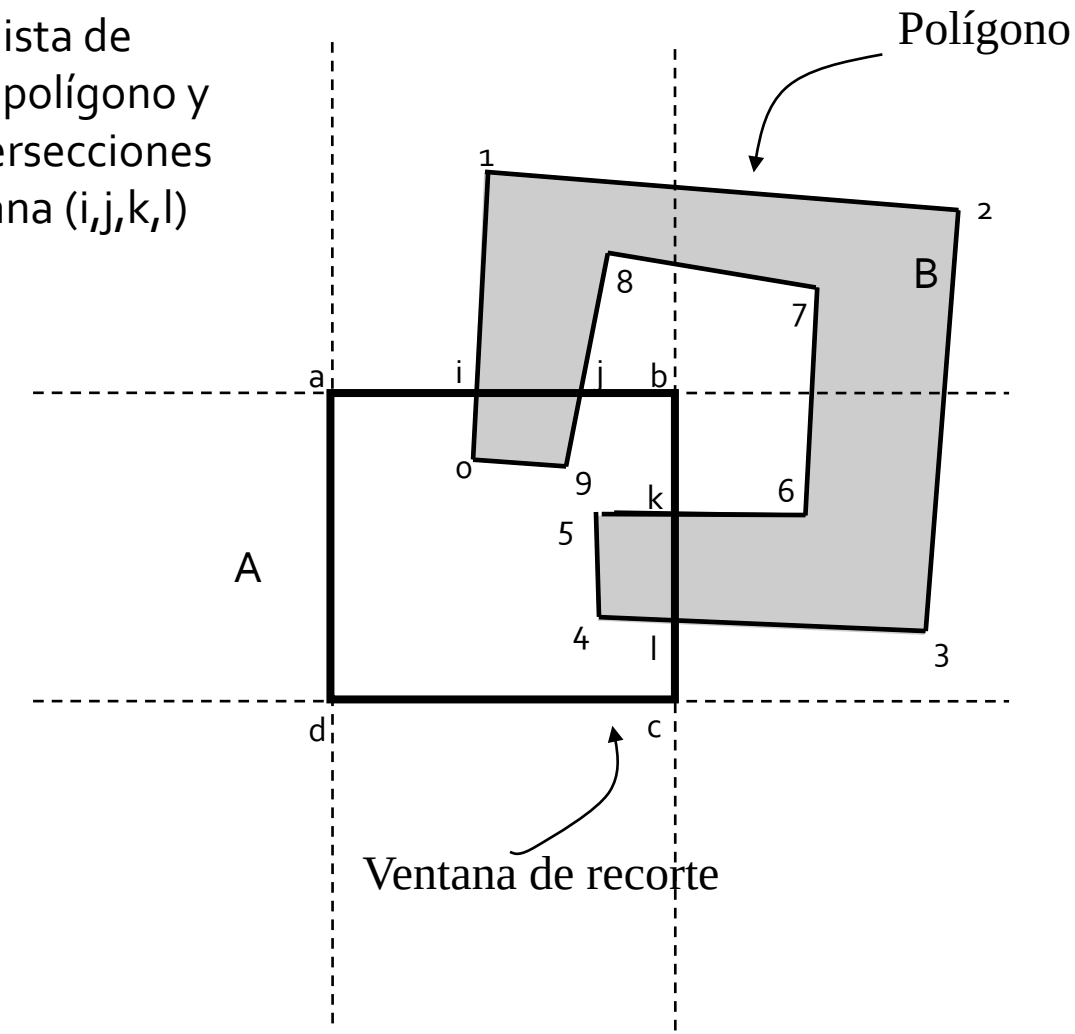
Recortado de polígonos

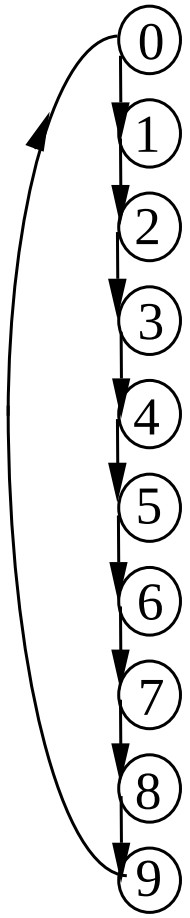
- Cuando los polígonos no son convexos, puede ocurrir que el resultado del recorte no sea un único polígono
- El algoritmo de Weiler-Atherton crea polígonos separados para cada fragmento visible

Recortado

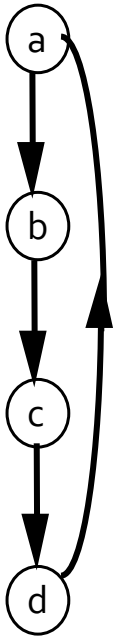
Recortado de polígonos

Recorrer la lista de vértices del polígono y calcular intersecciones con la ventana (i,j,k,l)





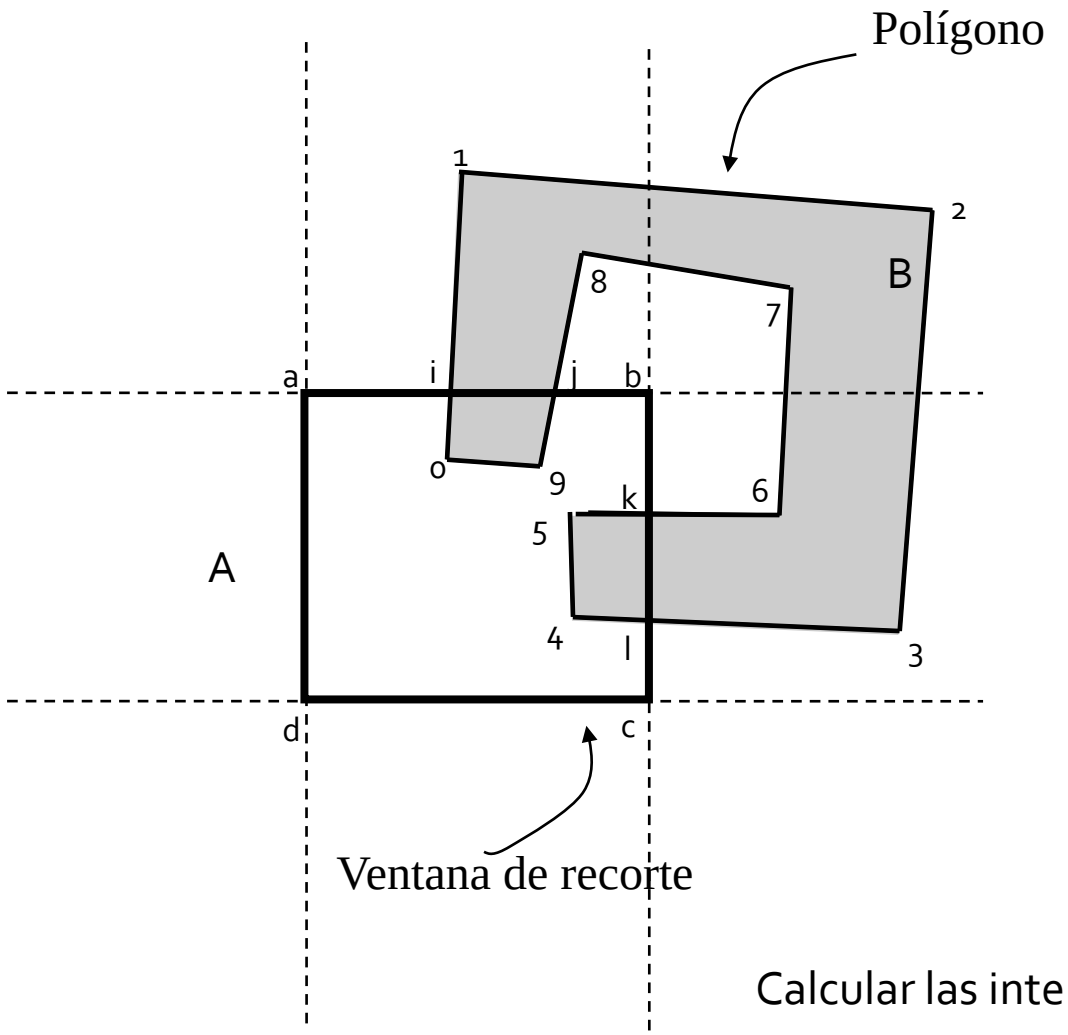
Lista de vértices del polígono

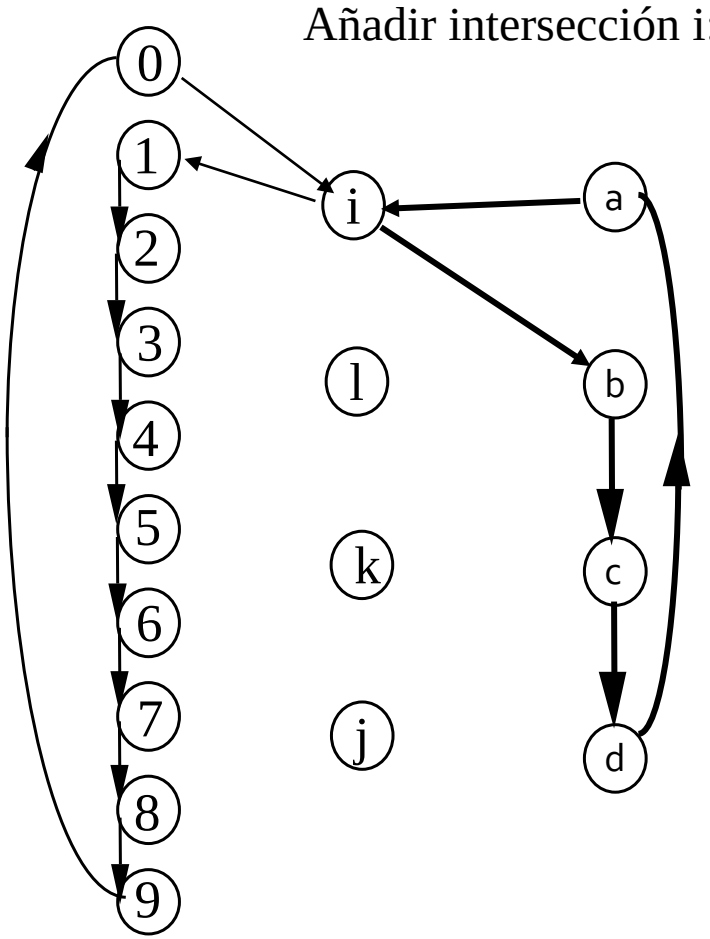


Lista de vertices de la ventana

Recortado

Recortado de polígonos

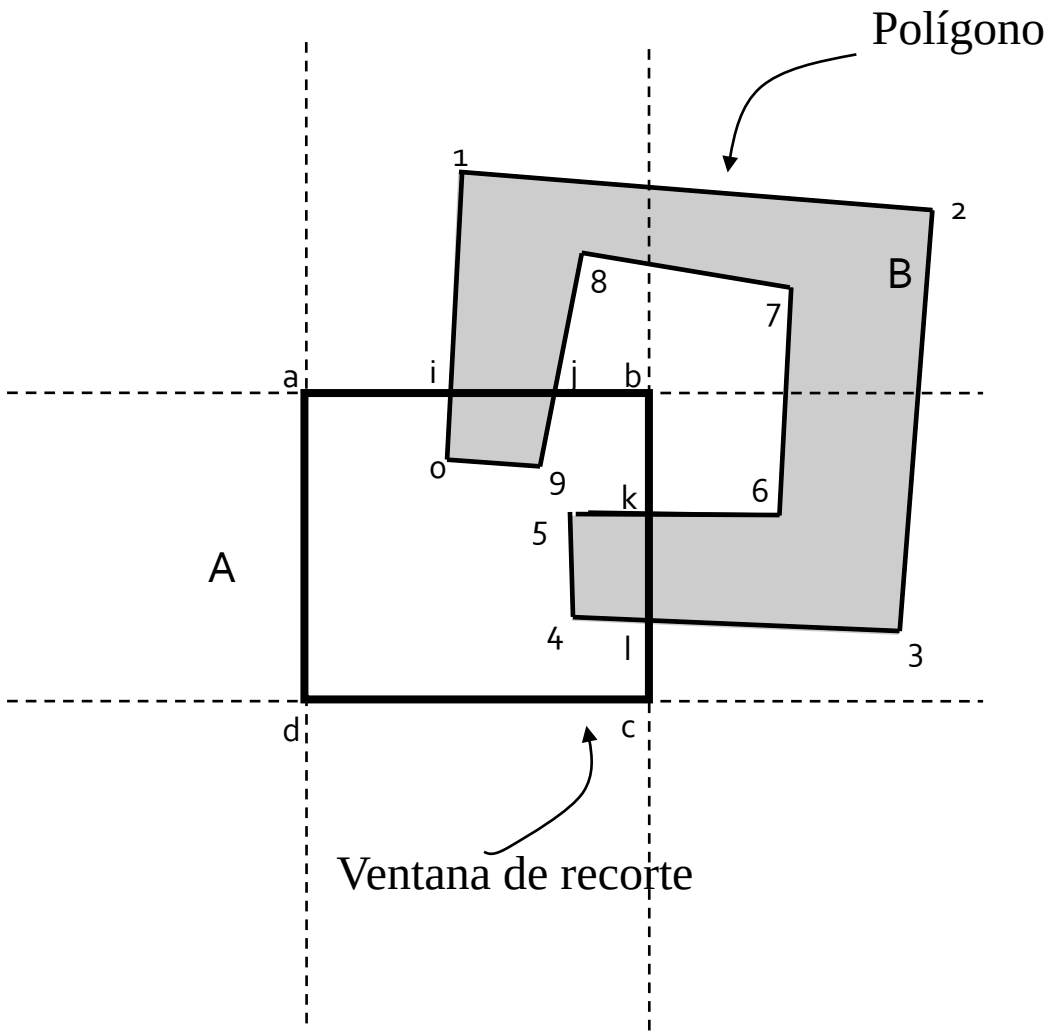


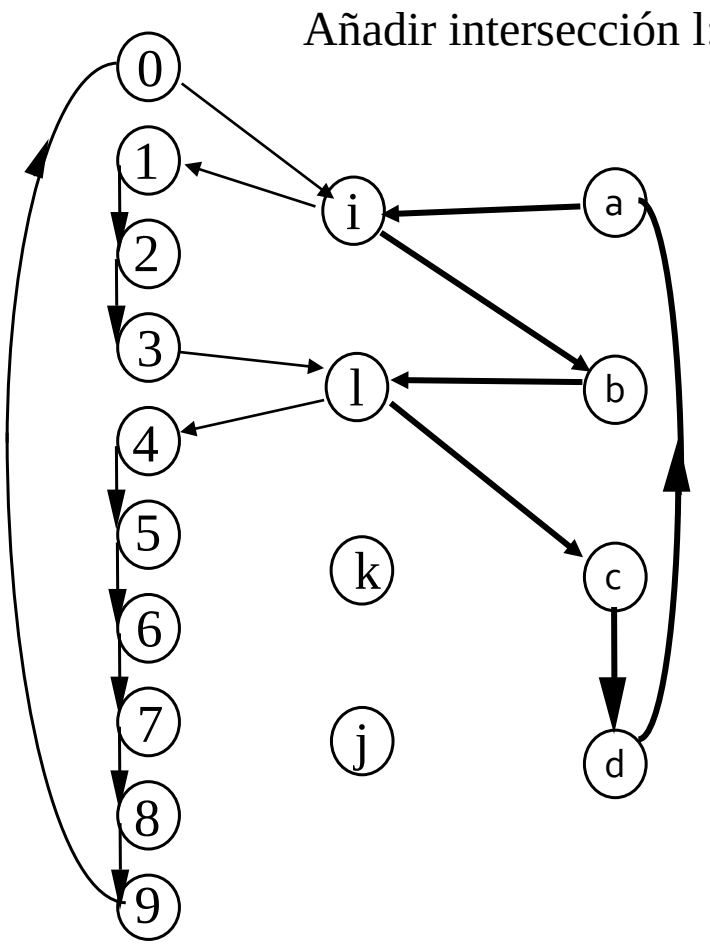


Calcular las intersecciones y conectarlas con las listas

Recortado

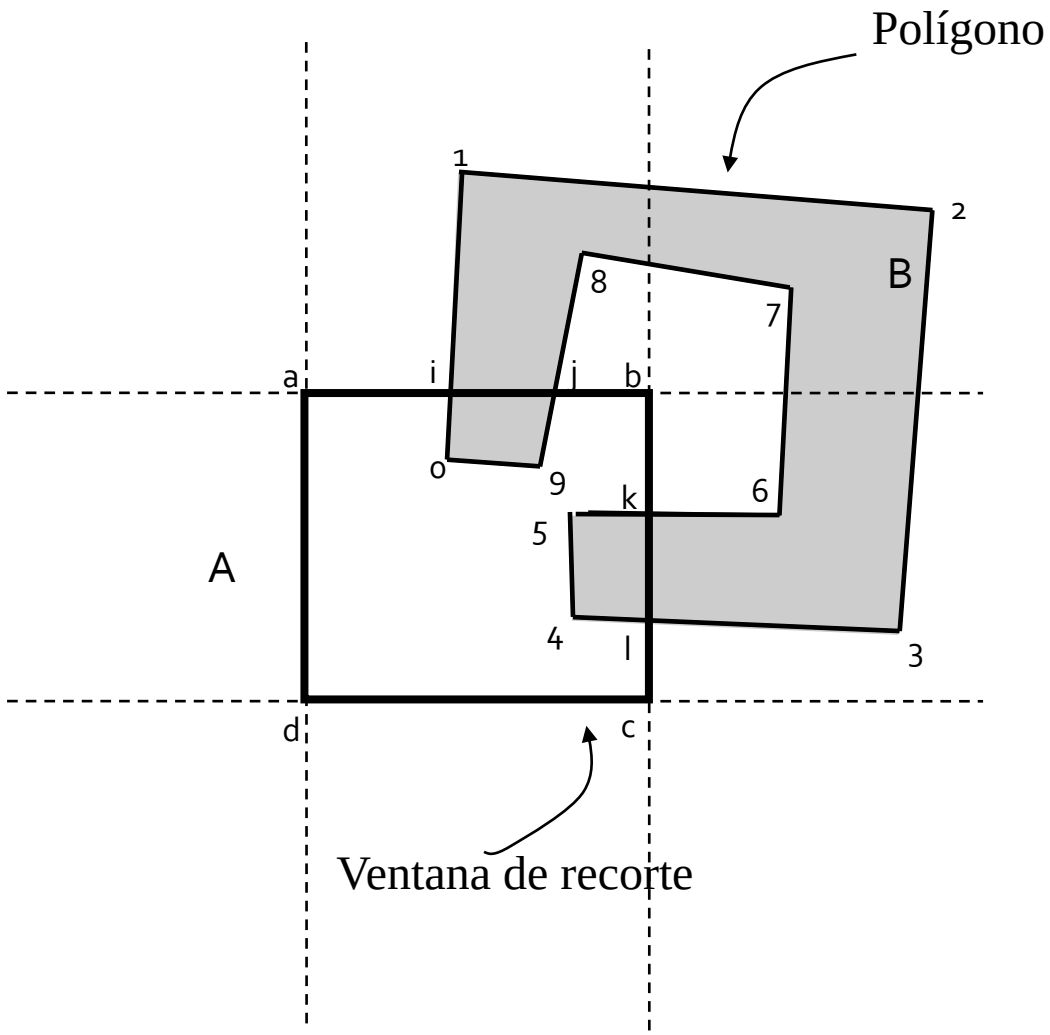
Recortado de polígonos

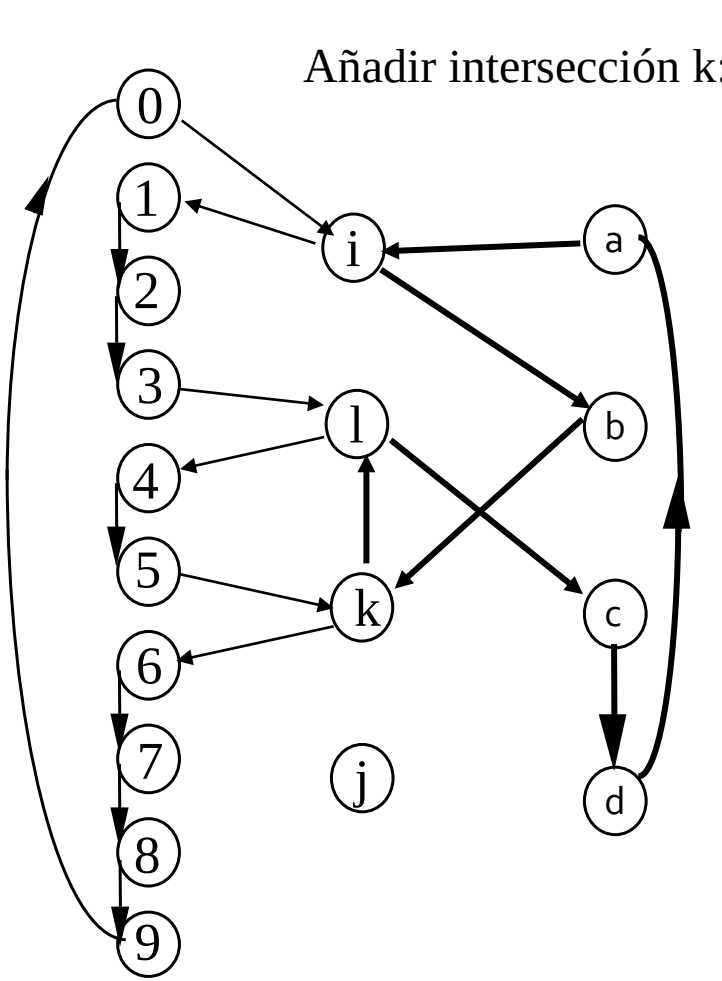




Recortado

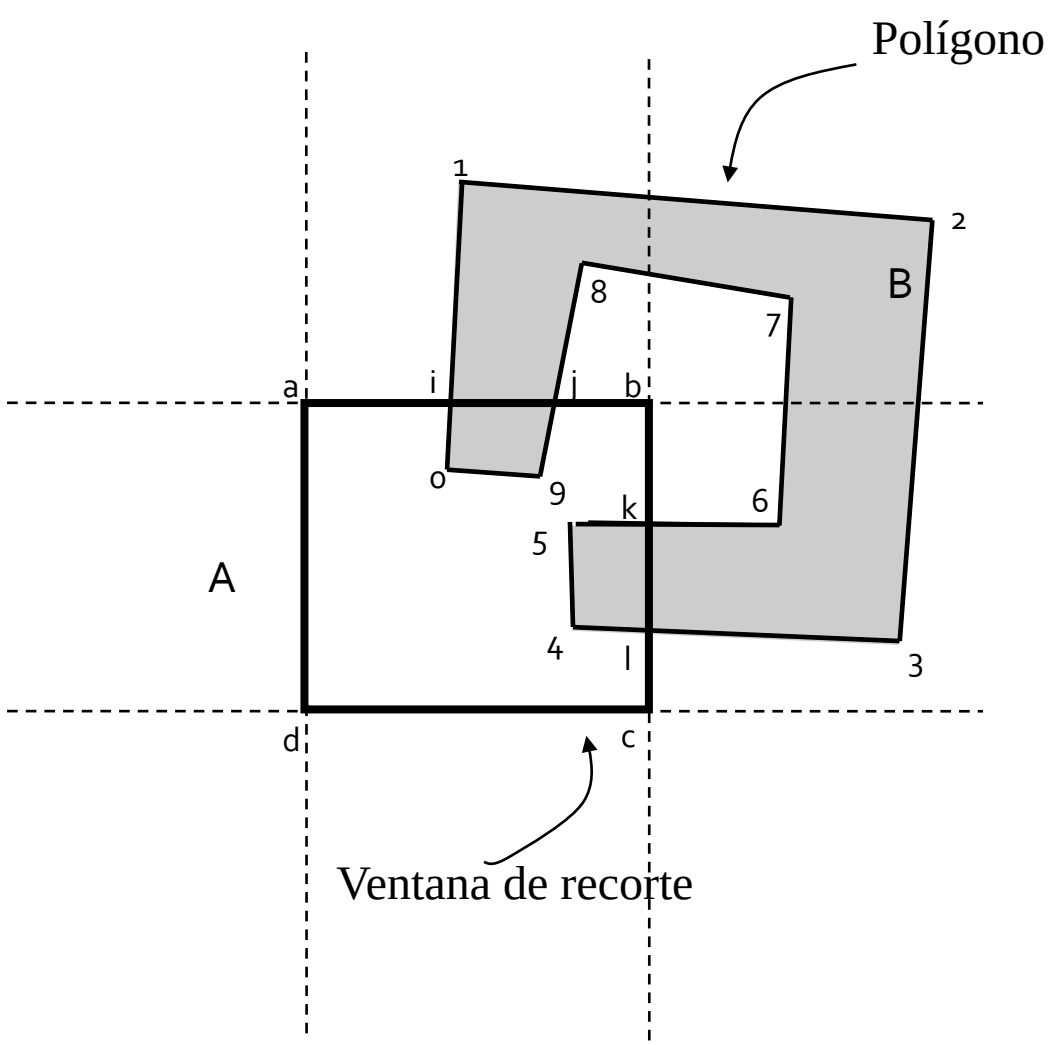
Recortado de polígonos

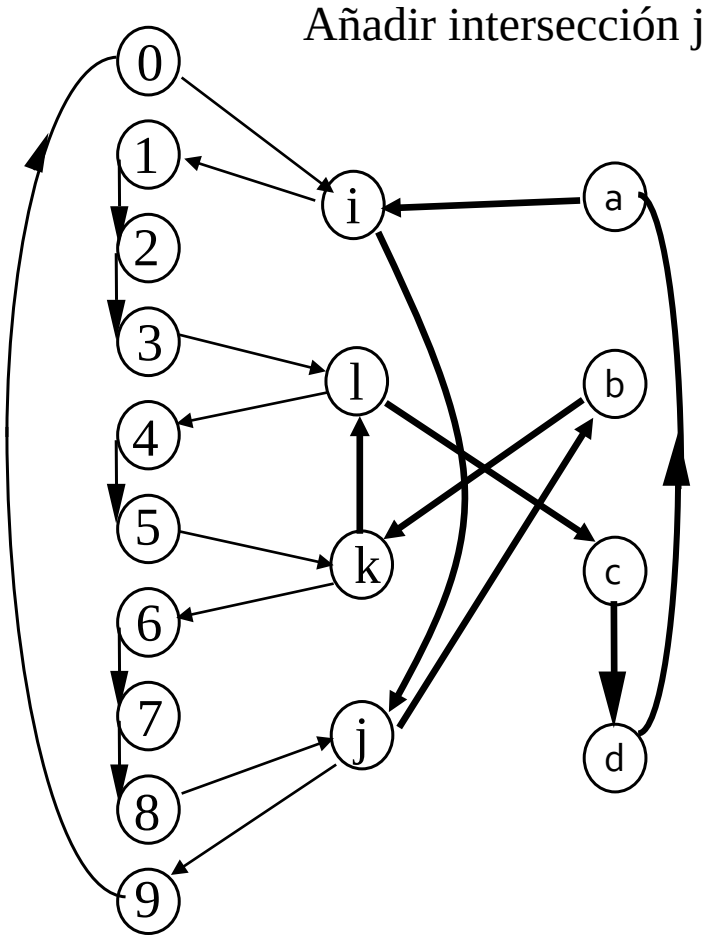




Recortado

Recortado de polígonos

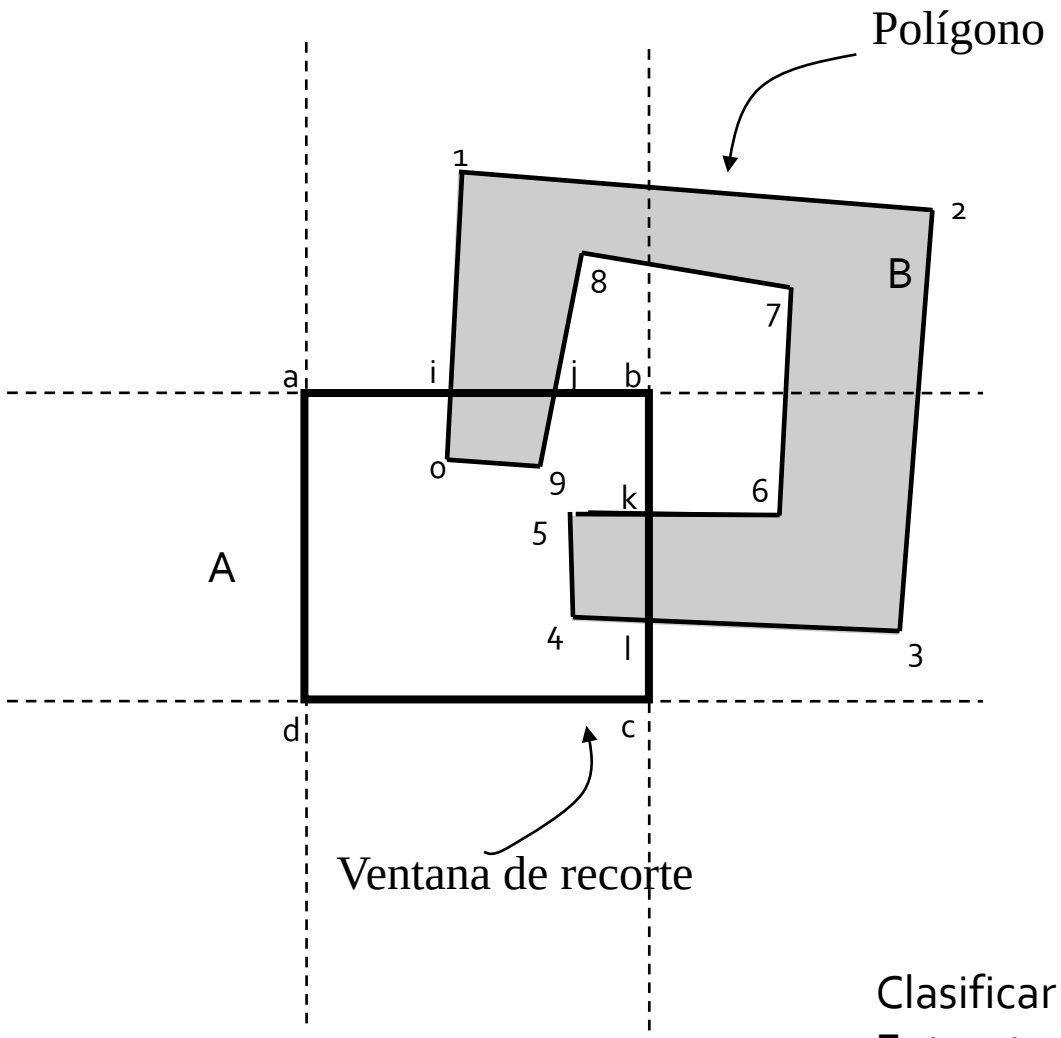


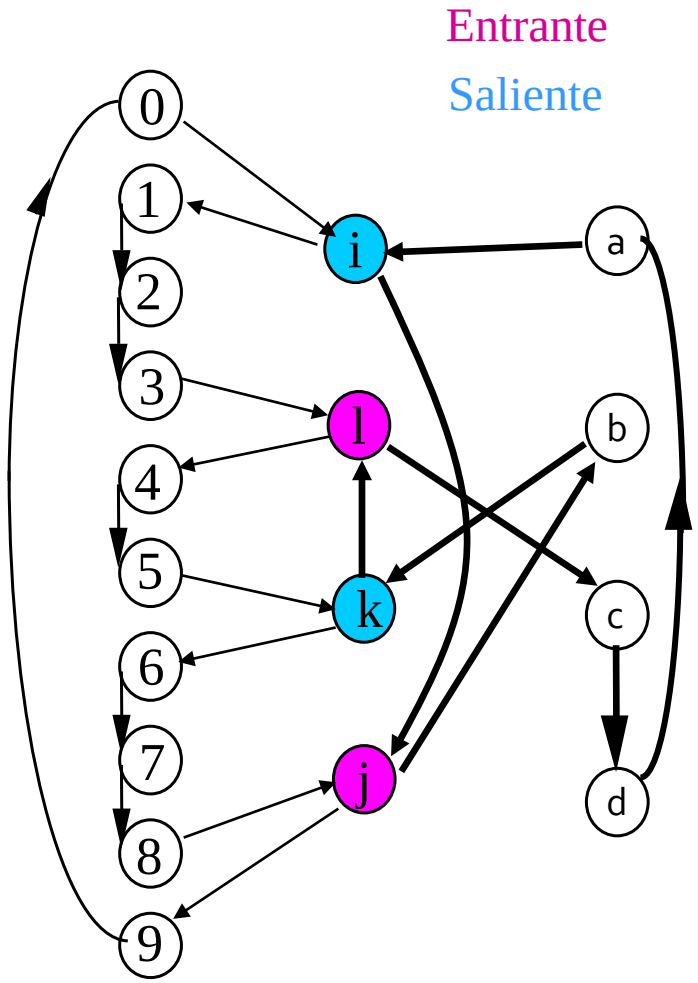


Bucle terminado

Recortado

Recortado de polígonos

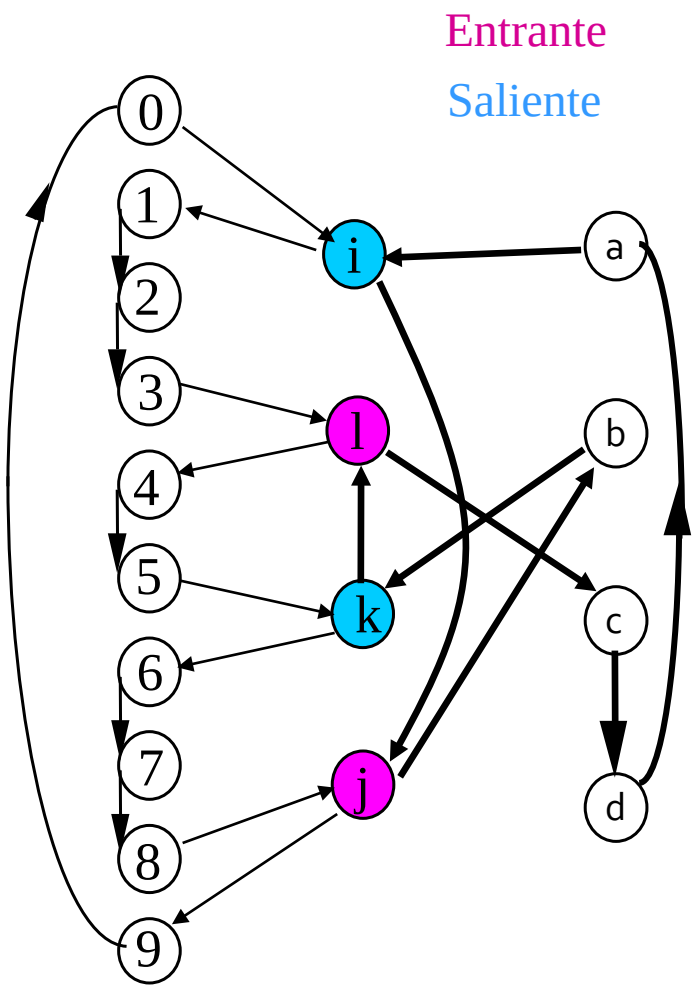




Clasificar cada intersección como:
Entrante o Saliente

Recortado

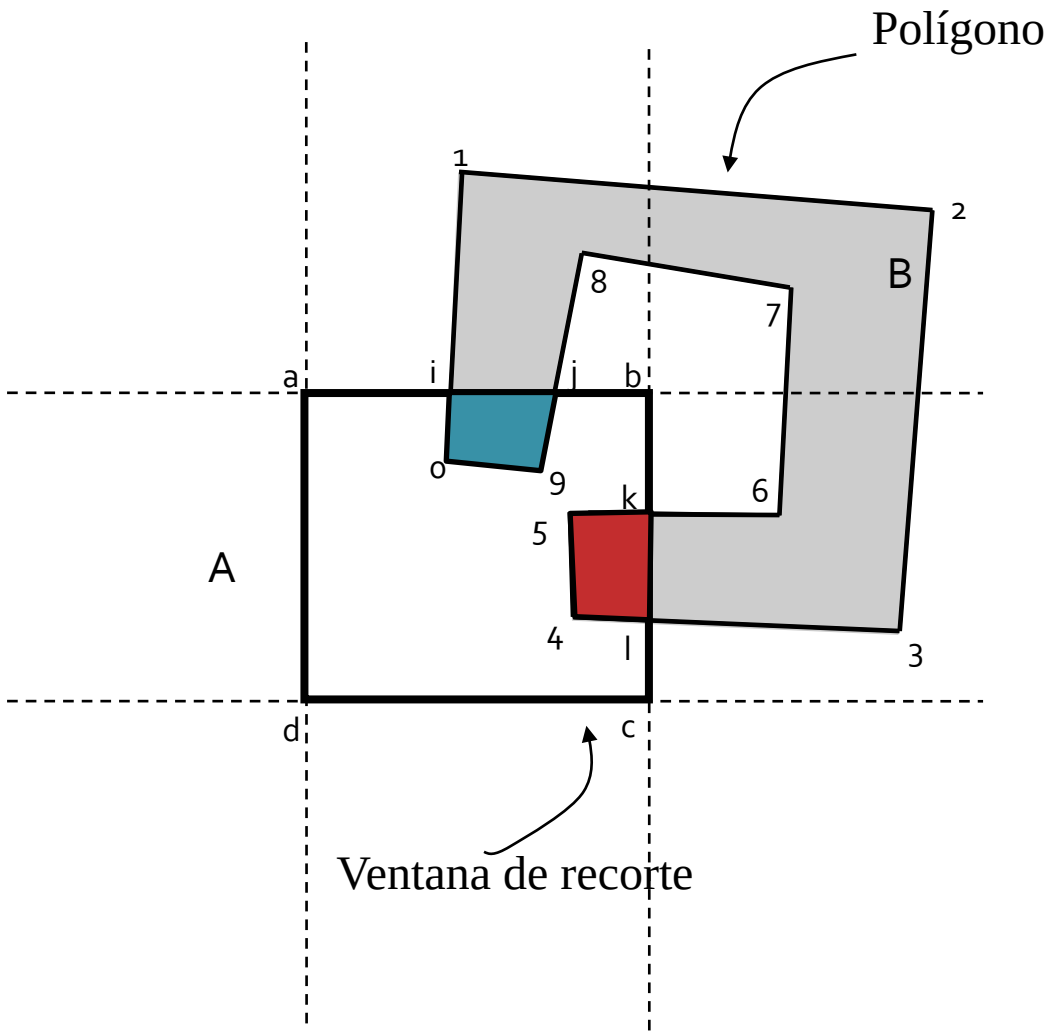
Recortado de polígonos

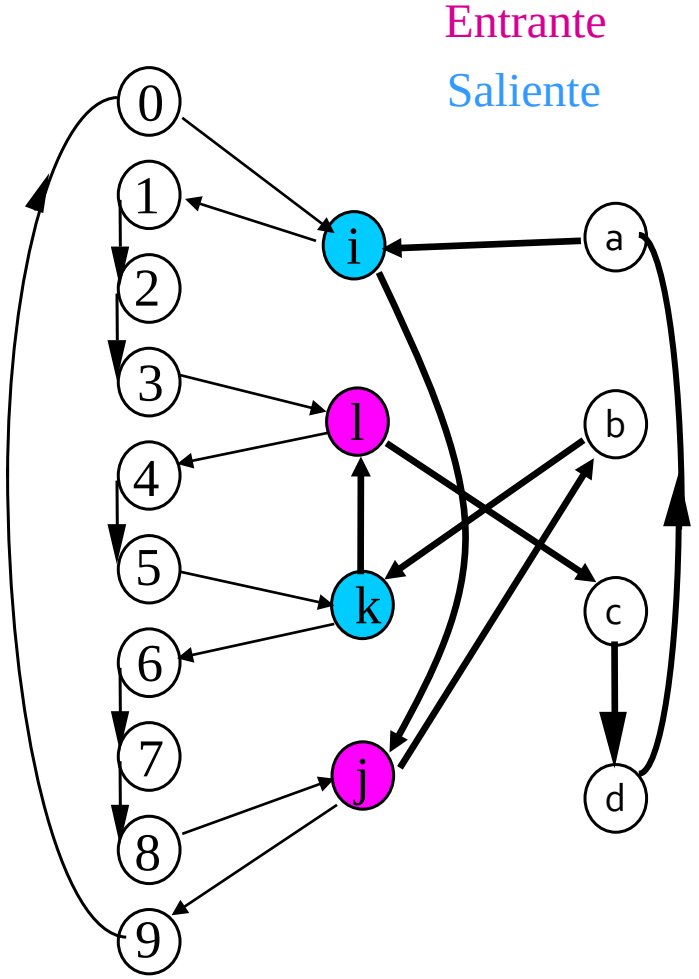


- Comenzar por una intersección entrante y recorrer la lista
- Si se encuentra un vértice saliente se sigue la lista de la región recorte
- Si se encuentra un vértice entrante se sigue la lista del polígono
- El polígono se cierra cuando llegas al primero.
- El algoritmo termina cuando se han recorrido todos los vértices entrantes

Recortado

Recortado de polígonos





- Polígono 1: L, 4, 5, K
- Polígono 2:J, 9, 0, I

Bibliografía

- D. Hearn, M. Baker. Computer Graphics with OpenGL. Pearson Prentice Hall, 4ª edición.
  - Apéndice A
  - Capítulos 7 y 8