Informática gráfica

Fernando Bobillo

Resumen de contenidos

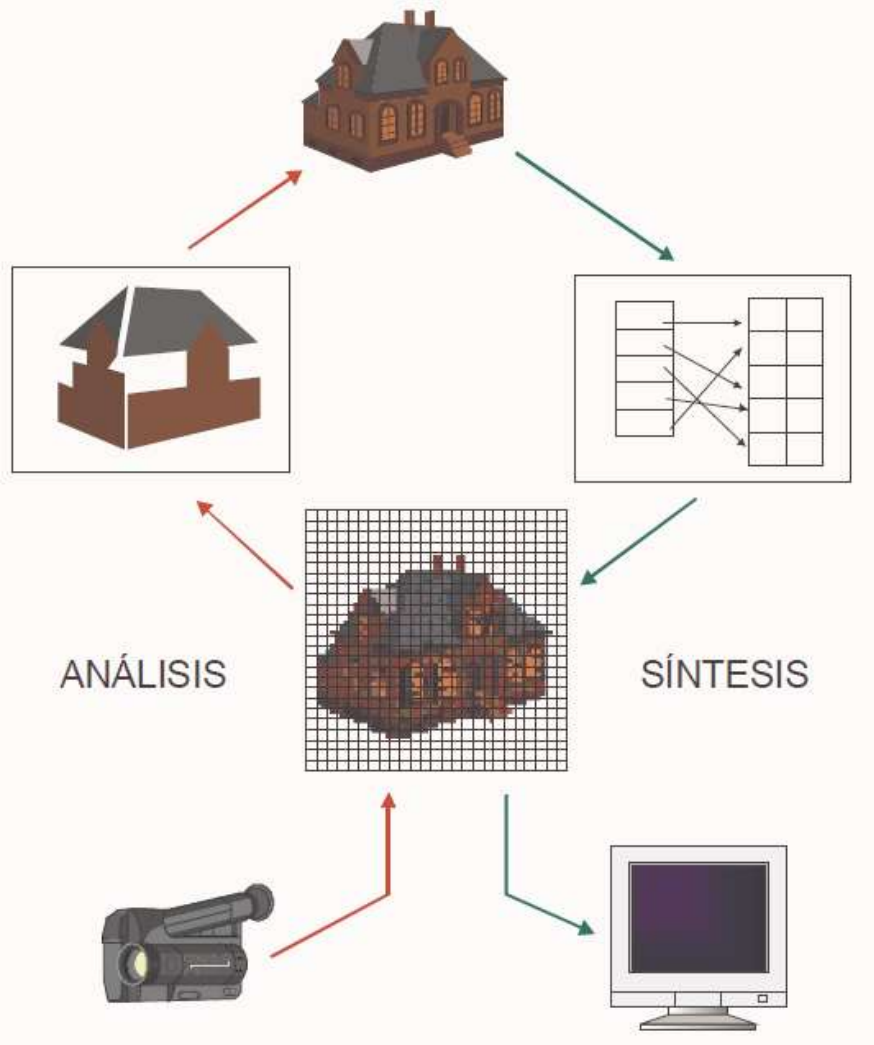
- Informática gráfica
- Modelos de representación
- Representación de color
- Modelado de objetos 3D
- Visualización de objetos 3D
- Visualización de objetos 2D
- Investigación en la Universidad de Zaragoza

Informática gráfica

Informática gráfica

- Gráficos por ordenador: creación, manipulación, análisis e interacción de las representaciones pictóricas de objetos usando ordenadores
- Informática gráfica: síntesis pictórica de objetos reales o imaginarios basada en sus modelos de ordenador
- El ordenador sirve para generar imágenes y son las propias imágenes el fin, no un medio
- Diferente a otros campos como el tratamiento de imágenes con un proceso contrario: analizar la escena o reconstruir modelos 2D ó 3D a partir de imágenes
- Importancia: la información más fácilmente asimilable por humanos, comunicación universal, compactación…

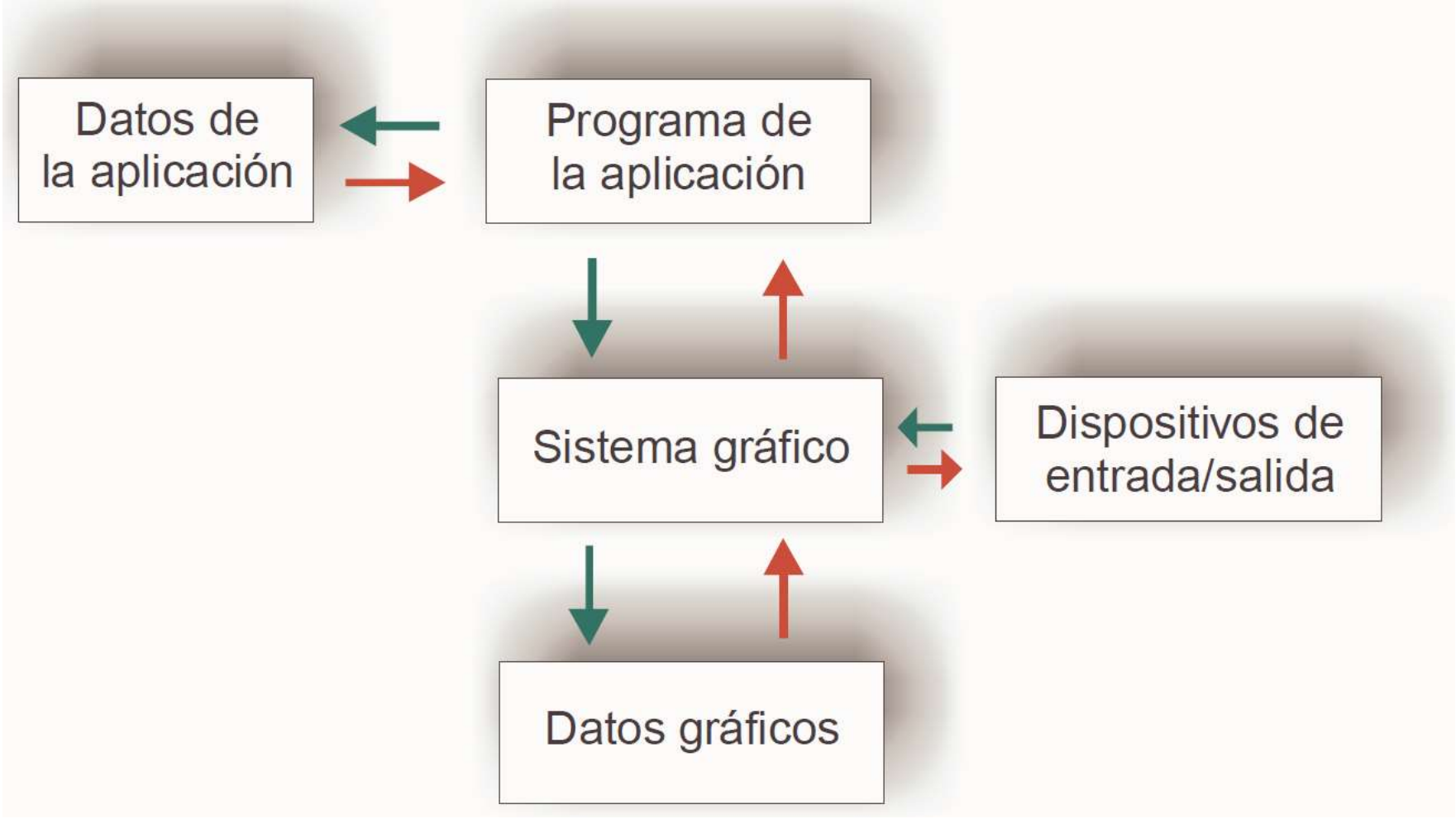
Informática gráfica



Ventajas del ordenador

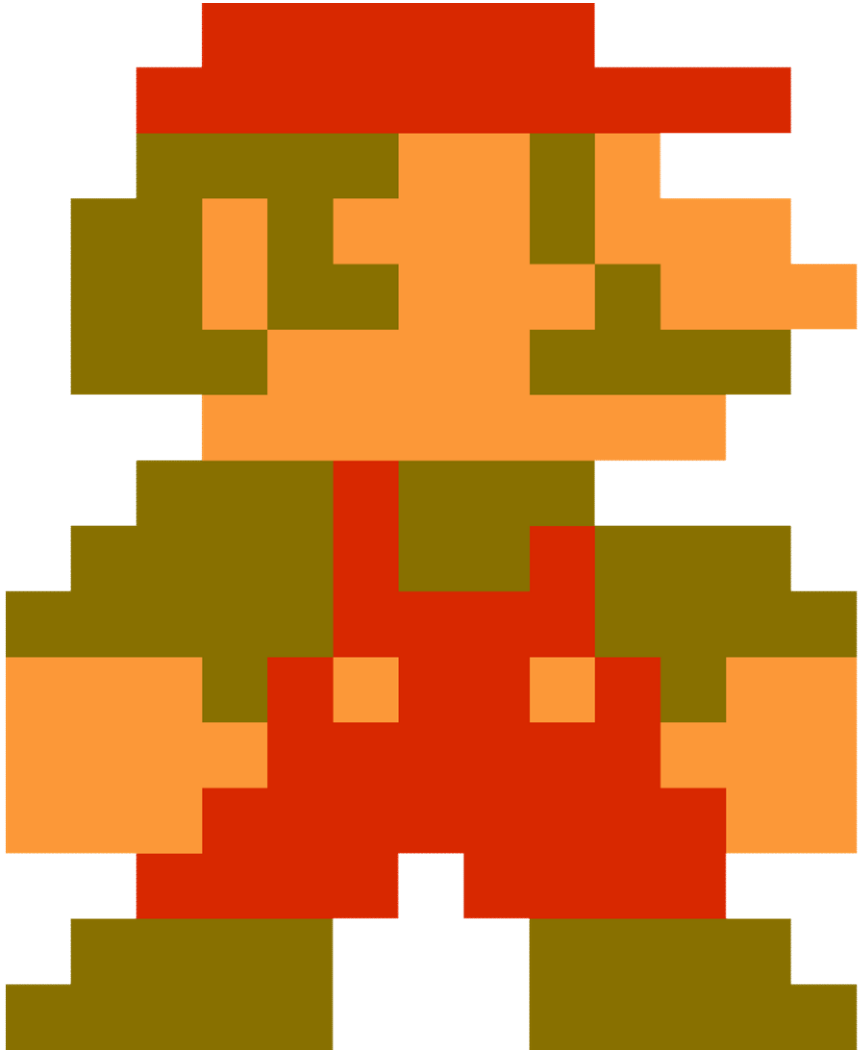
- Alto nivel
  - Frente a otros medios para producir imágenes (cámara, vídeo), permite generar y explorar mundos virtuales, con escalas iguales y diferentes a las humanas
- Bajo nivel
  - Almacenamiento de la información
  - Edición interactiva
  - Simulación de herramientas y efectos
- Aplicaciones
  - Diseño asistido por ordenador, ingeniería, arte, medicina, comercio, control de procesos, cartografía y sistemas de información geográfica, realidad virtual…

Modelo general de una aplicación gráfica



Modelos de representación

Píxeles



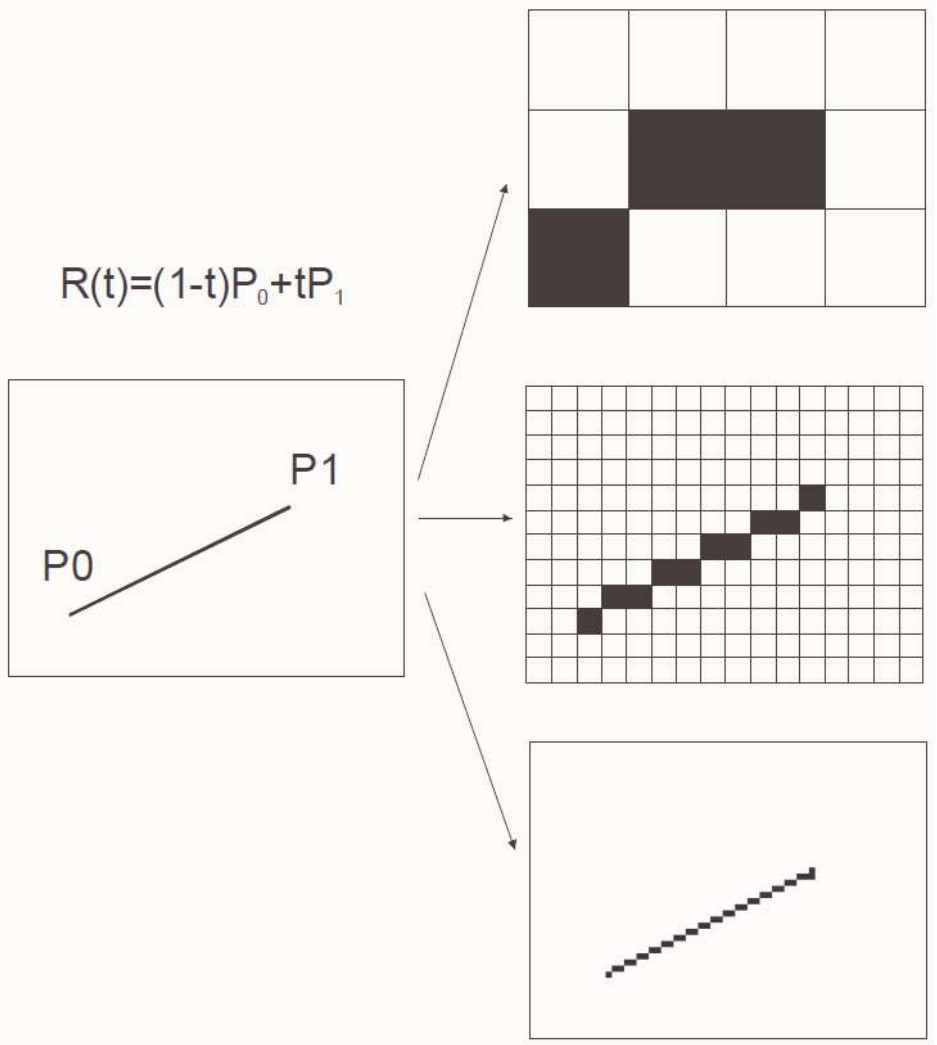
Píxel a píxel

Métodos de representación y creación

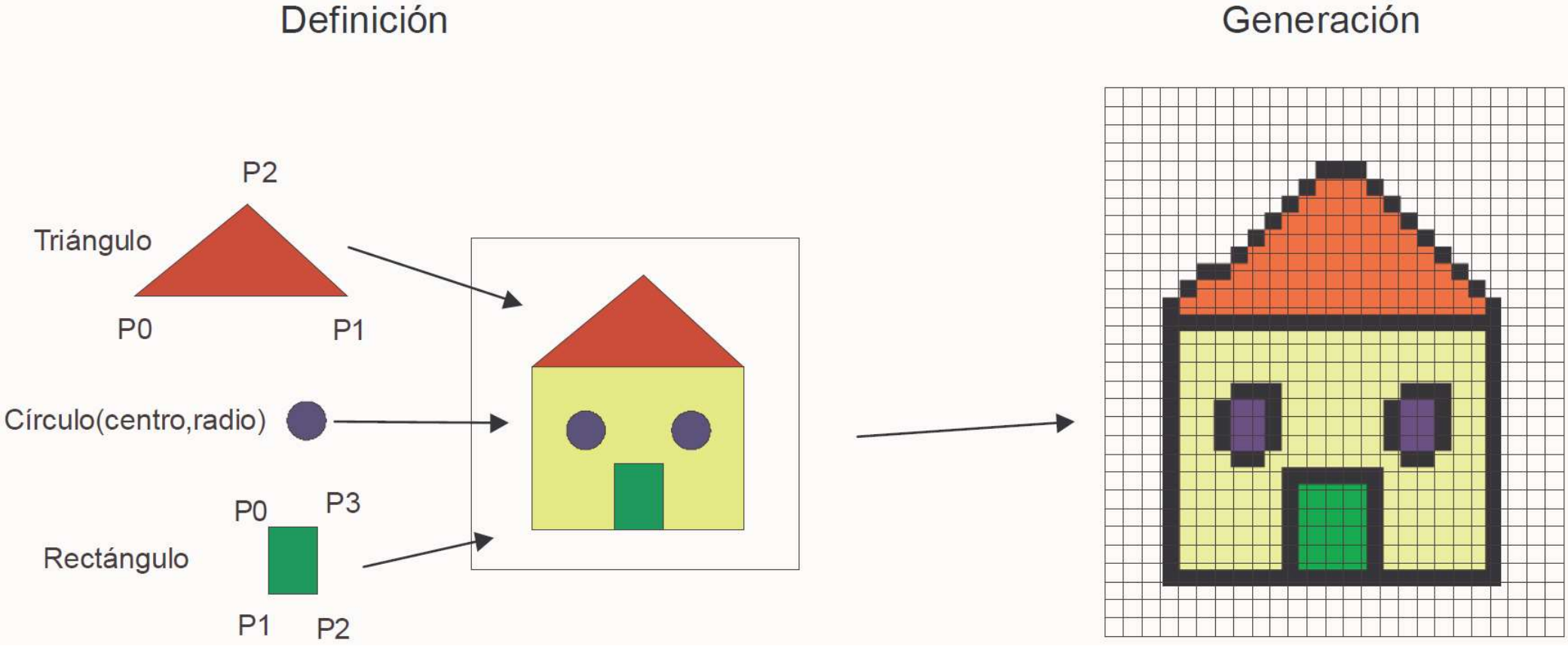
- Continua
  - Visión analógica: entre 2 elementos siempre existe otro
  - Formulación matemática
  - Resolución infinita
  - Ejemplo: formato svg
  - Representación vectorial
- Discreta
  - Visión digital: elementos totalmente separados
  - Imagen: conjunto finito de áreas (píxeles), cada una de un color
  - Resolución finita
  - Ejemplo: formato jpg
  - Representación matricial

Métodos de representación y creación

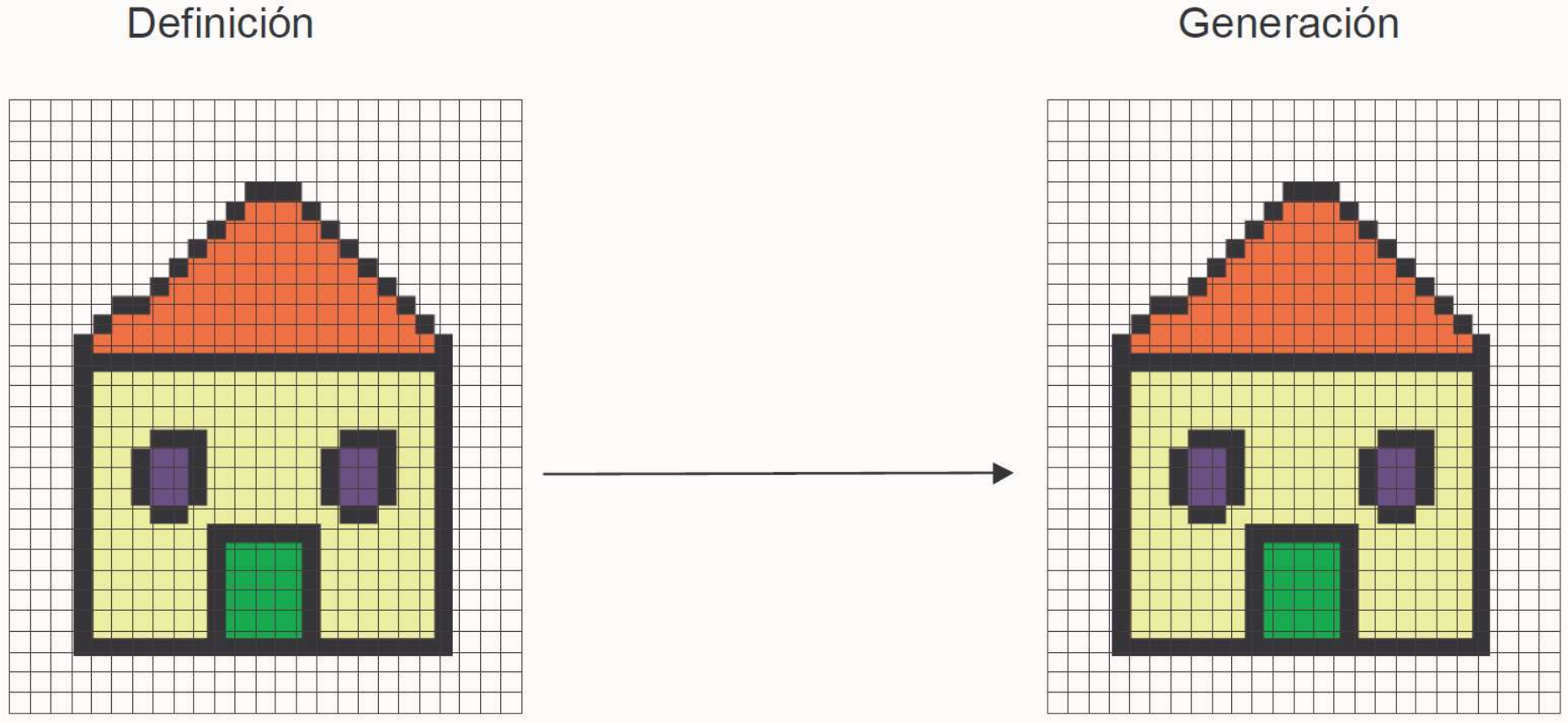
- Hay que diferenciar dos conceptos diferentes
- Definición del modelo:
  - Conjunto de primitivas gráficas o elementos gráficos que indican composición, forma, color de una entidad
- Generación de la imagen:
  - Conjunto de píxeles que forman la imagen del modelo



Método vectorial



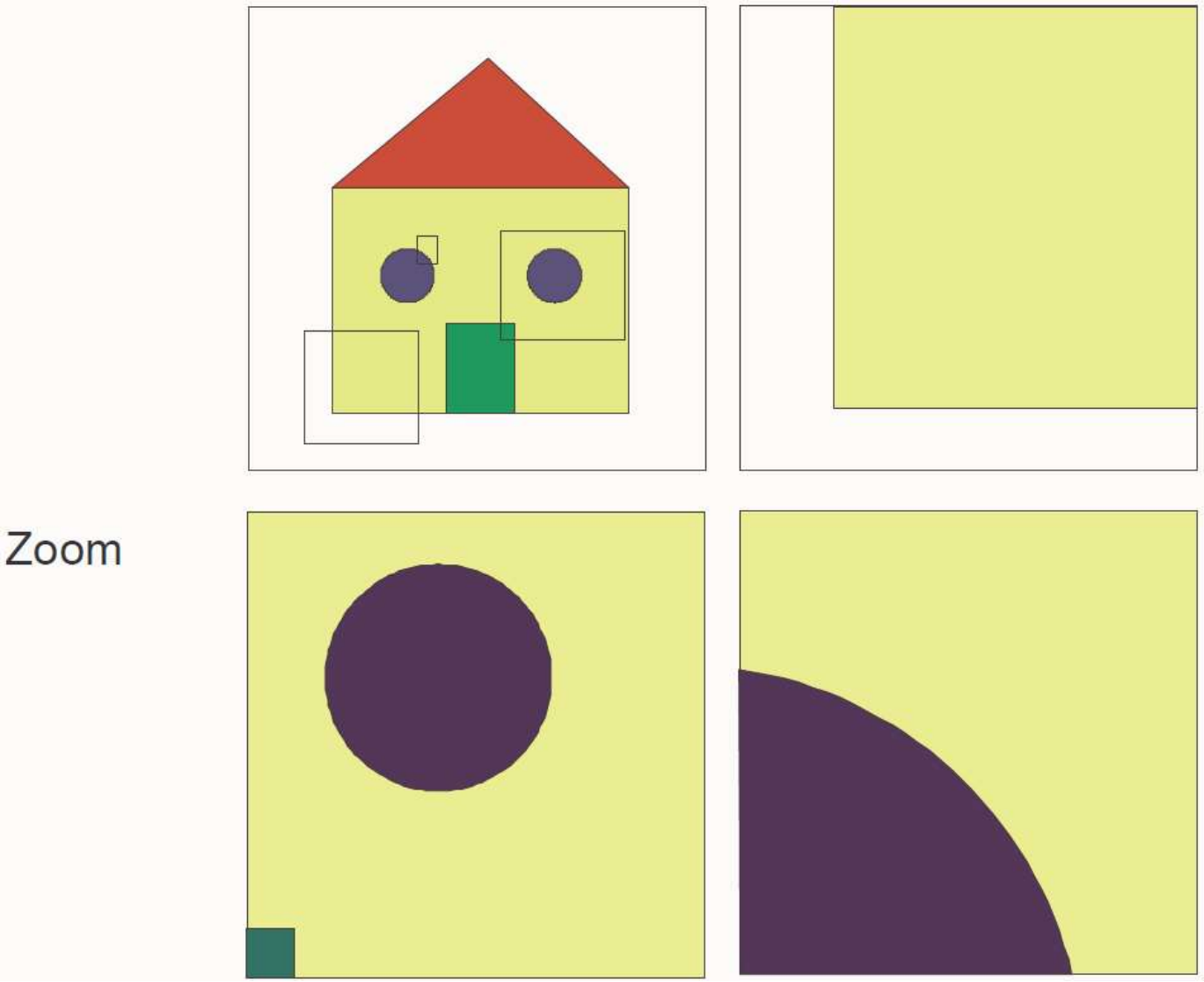
Método matricial



Métodos de representación y creación

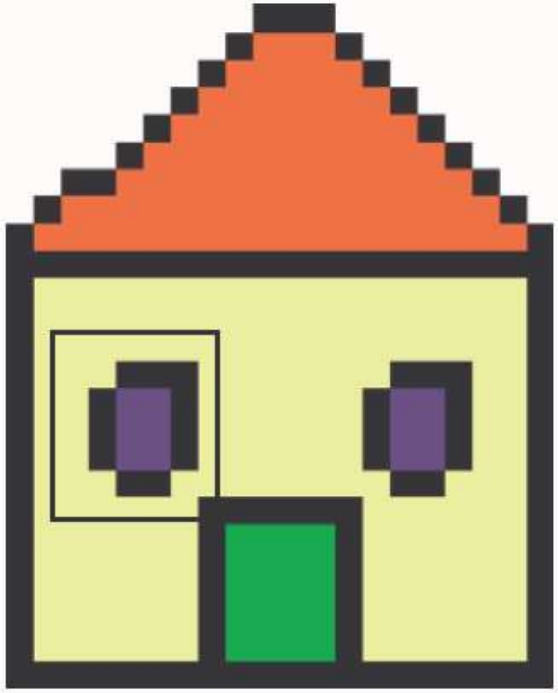
- Representación vectorial
  - – Sistema de coordenadas reales
  - ✓ Especificación de alto nivel, muy compacto
  - ✓ Resolución infinita (infinitos niveles de zoom)
  - ✓ Escala variable (ajustable por el usuario)
  - ✓ Espacio y precisión infinitos
  - – En teoría, en la práctica fijados por la precisión del ordenador
- Representación matricial
  - – Sistema de coordenadas enteras (píxeles)
  - ✓ Fácil de tratar al trabajar directamente con los píxeles
  - ✗ Espacio y precisión finitos
  - – Para aumentar el zoom a partir del máximo se debe interpolar

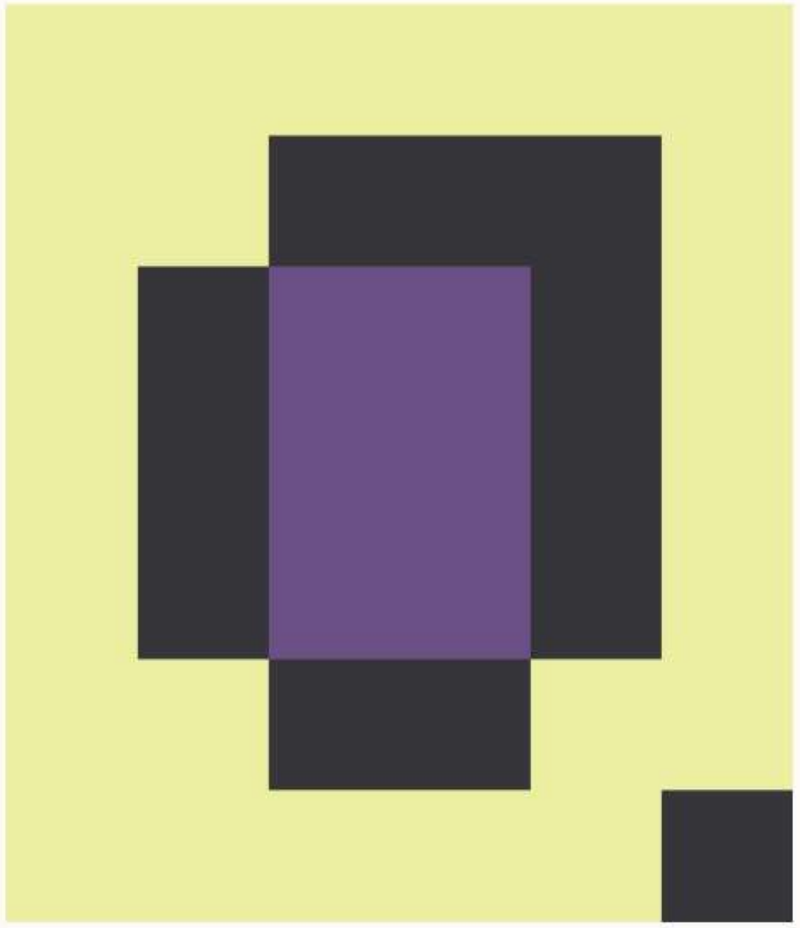
Método vectorial



Método matricial

Zoom

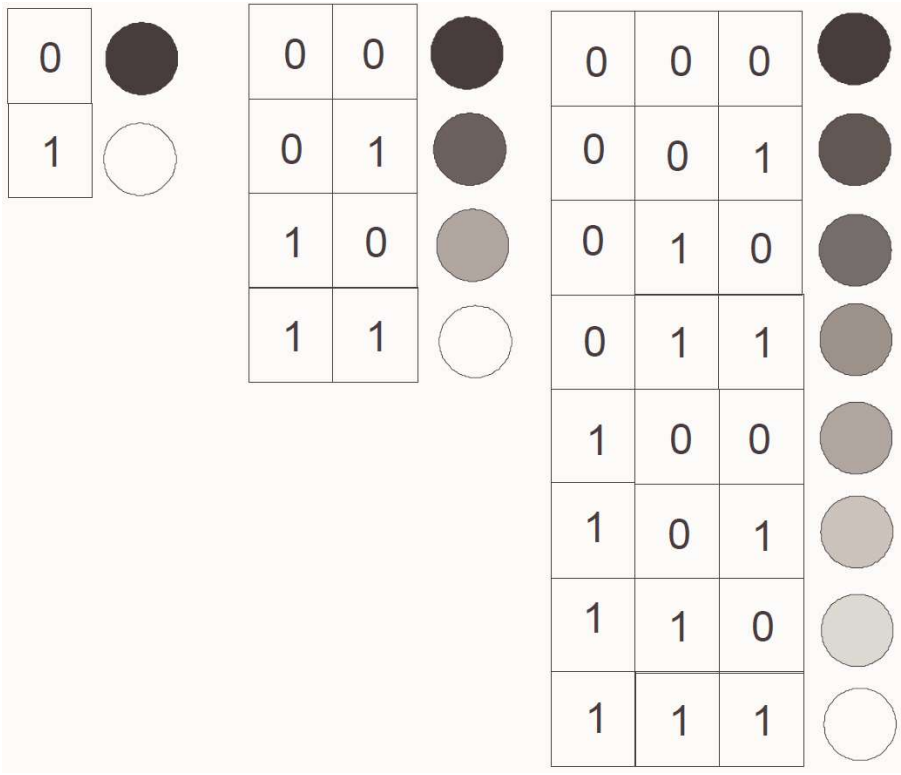




Representación de color

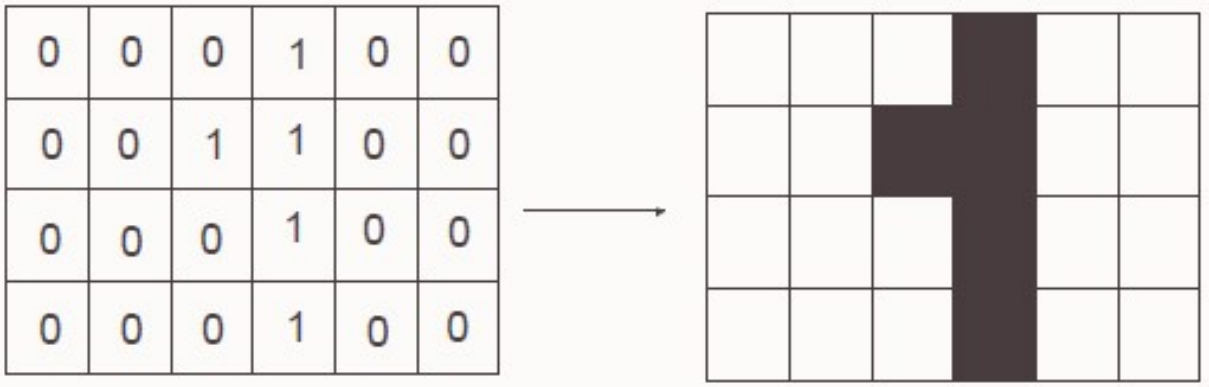
Representación de color

- Con n bits, podemos representar 2^n colores
- Normalmente, 1B por color primario (256 colores)
- Con 24 bits, 16 millones: color real o verdadero (el ojo humano no puede distinguir más colores)



Representación del color

- Los dispositivos de salida suelen usar tecnología matricial
- Una imagen es una matriz de píxeles
- Cada pixel posee un color
- Un color se describe mediante un valor numérico que codifica una intensidad
- Para un ordenador, una imagen es un conjunto de números almacenados en memoria

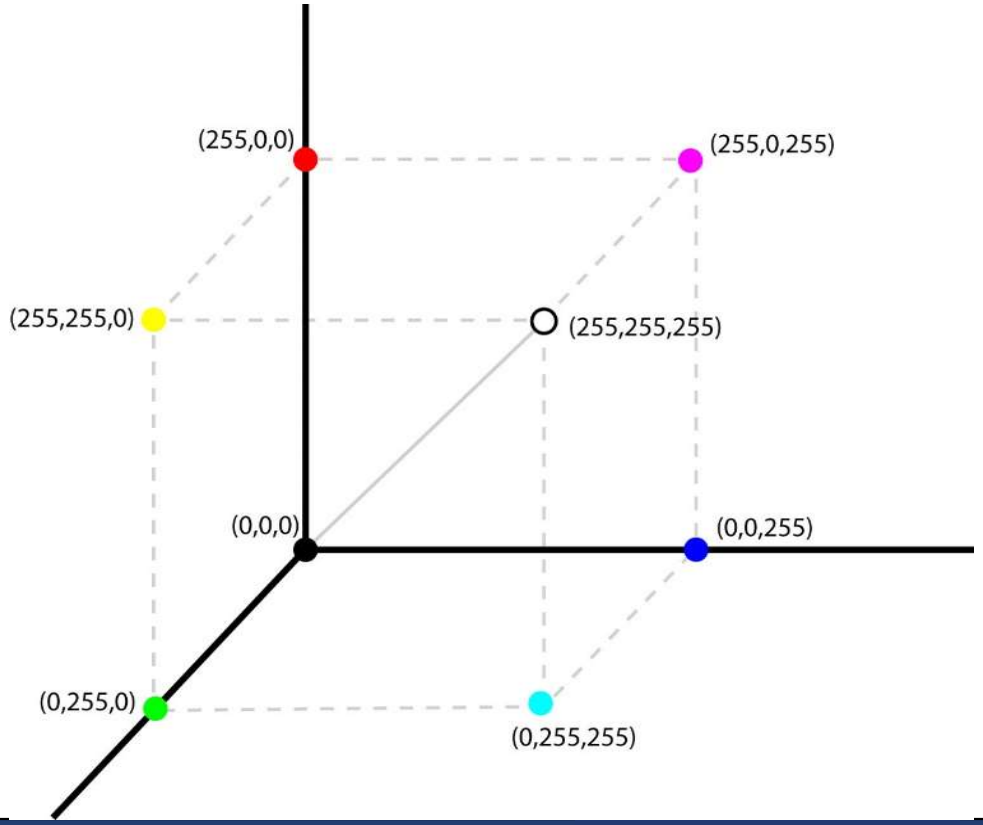


Representación del color

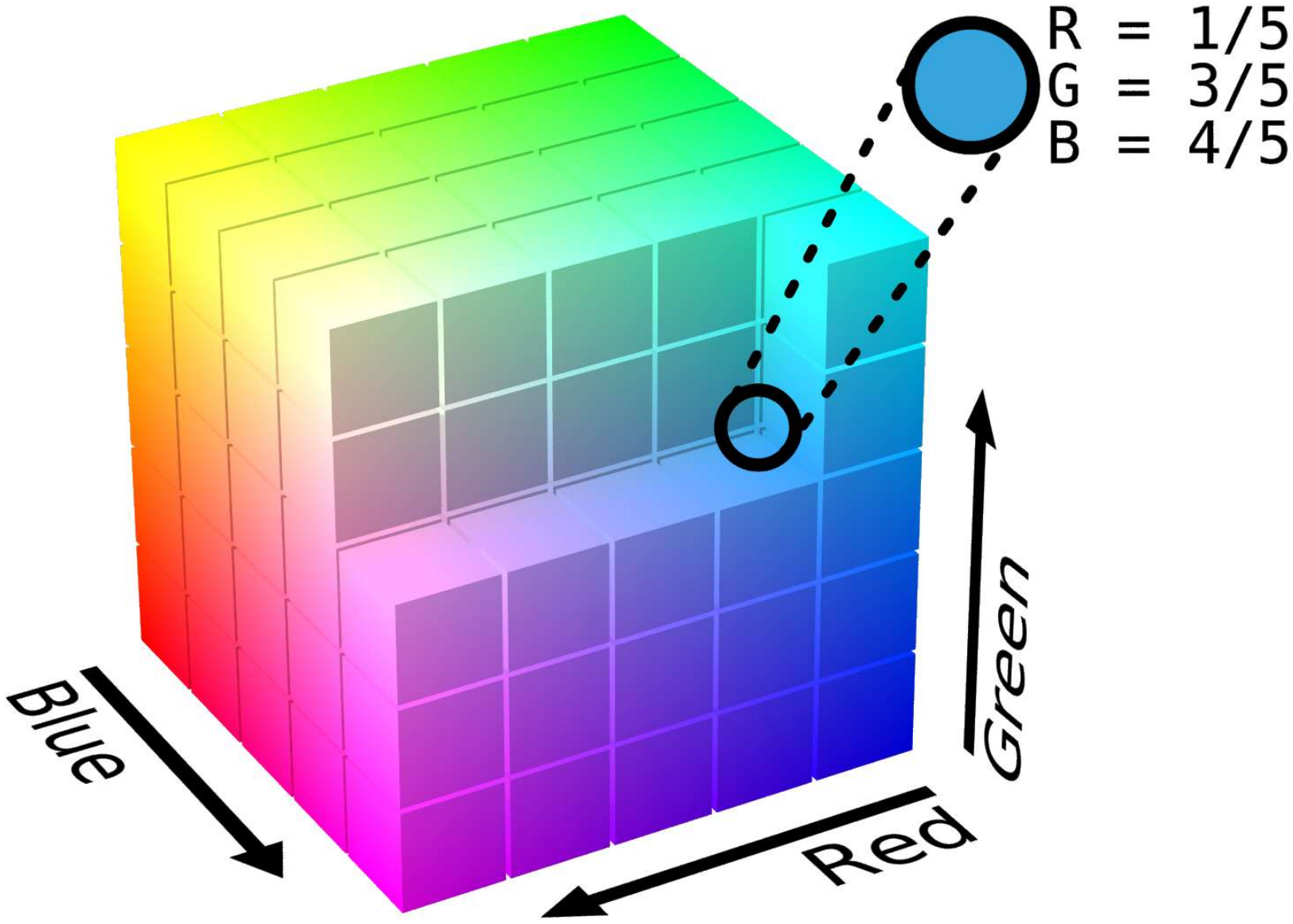
- Modelos orientados al hardware
  - Color descrito según el hardware que produce el color
  - Suelen ser difíciles de usar por los humanos
- Modelos orientados al usuario
  - Basado en la forma de operar de humanos (pintores…)
  - Se deben transformar a un modelo próximo al hardware
- Modelos aditivos
  - Se emite luz de una fuente
  - Habituales en televisores, monitores…
- Modelos sustractivos
  - El color de un objeto depende de las longitudes de onda que no absorbe, no de la luz que emite
  - Habituales en impresión, pintura, fotografía…

Representación de color: RGB

- Modelo RGB (Rojo, Verde, Azul)
- Modelo aditivo y orientado al hardware
- Se añaden colores básicos sobre fondo negro

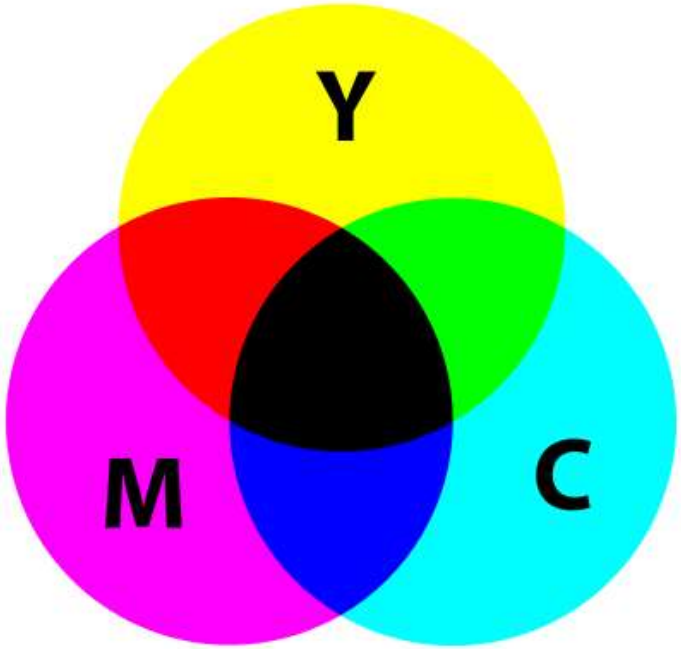


Representación de color: RGB



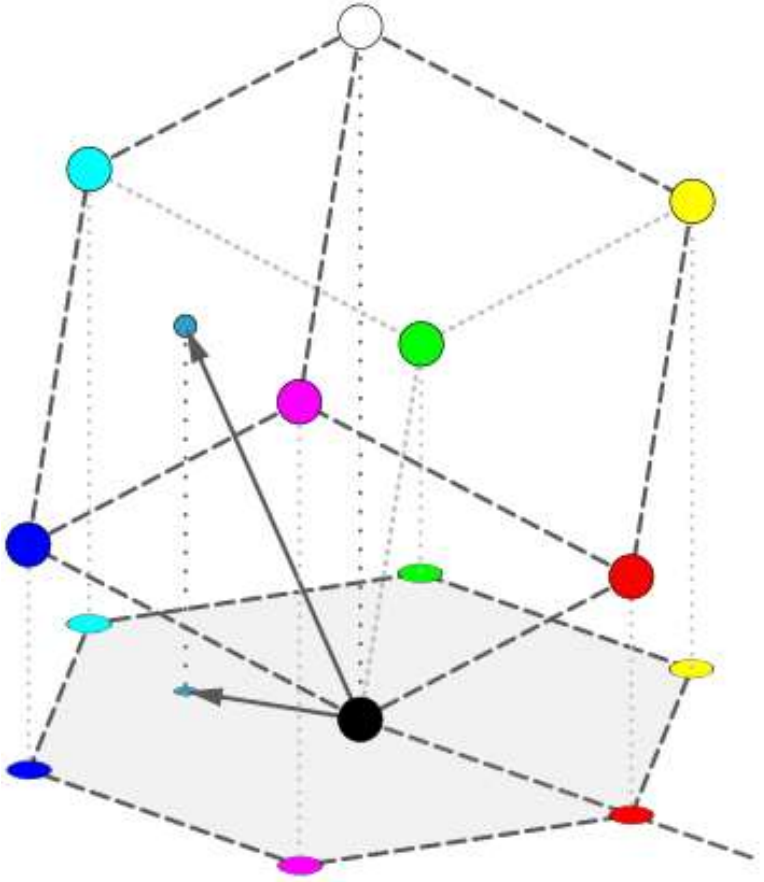
Representación de color: CMY

- Modelo CMY (Cián, Magenta y Amarillo)
- Modelo sustractivo y orientado al hardware
- Se añaden colores básicos sobre fondo blanco
- En la práctica, CMYK: Cián, Magenta, Amarillo y Negro

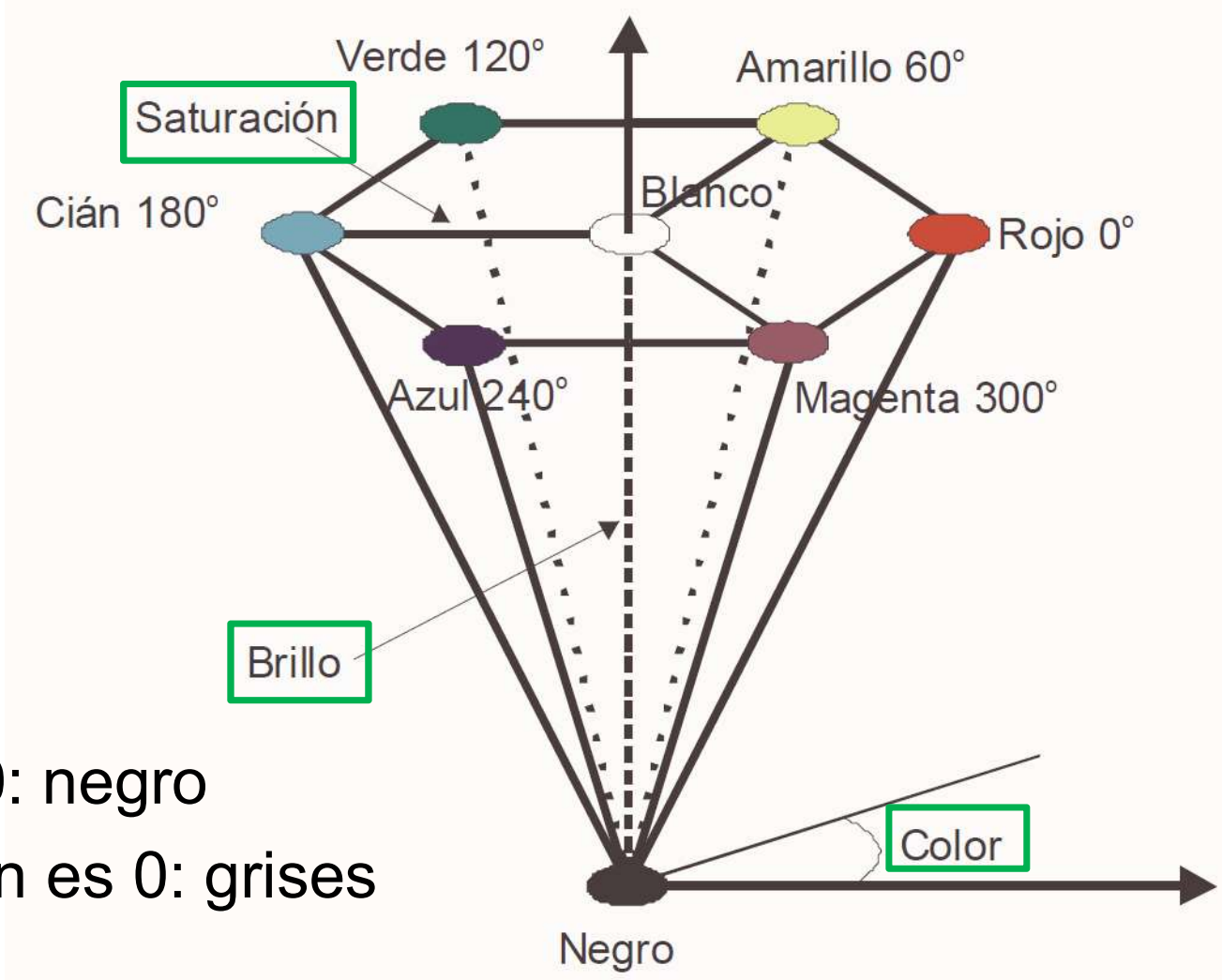


Representación de color: HSV

- Modelo HSV (Color, Saturación y Brillo)
- Orientado al usuario, como los pintores
  - Color + blanco: tonos pastel
  - Color + negro: tonos oscuros
- Espacio de definición: pirámide hexagonal
  - Plano cromático: perpendicular a la diagonal negro-blanco del cubo RGB
  - Al proyectar el cubo RGB en el plano cromático se obtiene un hexágono
- Si el brillo es 0, color y saturación son irrelevantes: color negro
- Si no, si la saturación es 0, el color es irrelevante: colores grises



Representación de color: HSV



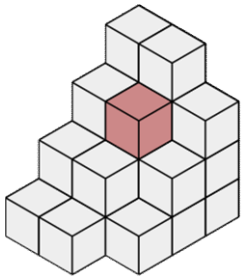
Si brillo es 0: negro

Si saturación es 0: grises

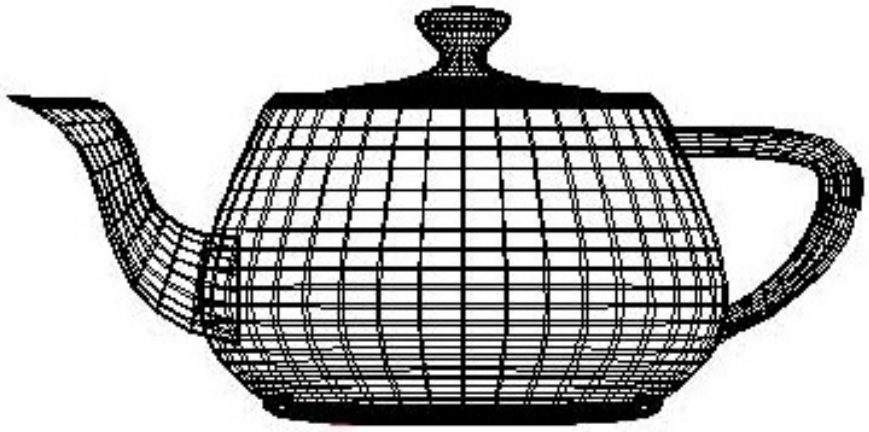
Modelado de objetos 3D

Representación de objetos

- Objetivo: modelizar objetos de la realidad tridimensional
- Modelo más sencillo: geometría
  - Vértices
- Modelos de alambres o mallas: geometría y topología
  - Vértices y aristas
- Modelos de superficie: interacción de la luz con el superficie
  - Vértices y caras; vértices, aristas y caras…
- Modelos de volumen: para ver el interior
  - Vóxel: elemento mínimo de volumen, generaliza el píxel



Modelos de alambres y superficies

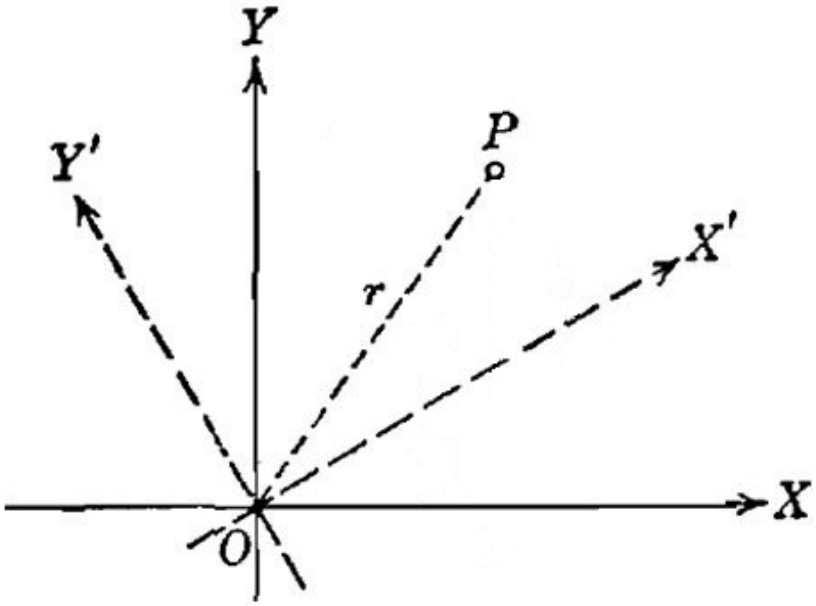


Transformaciones geométricas

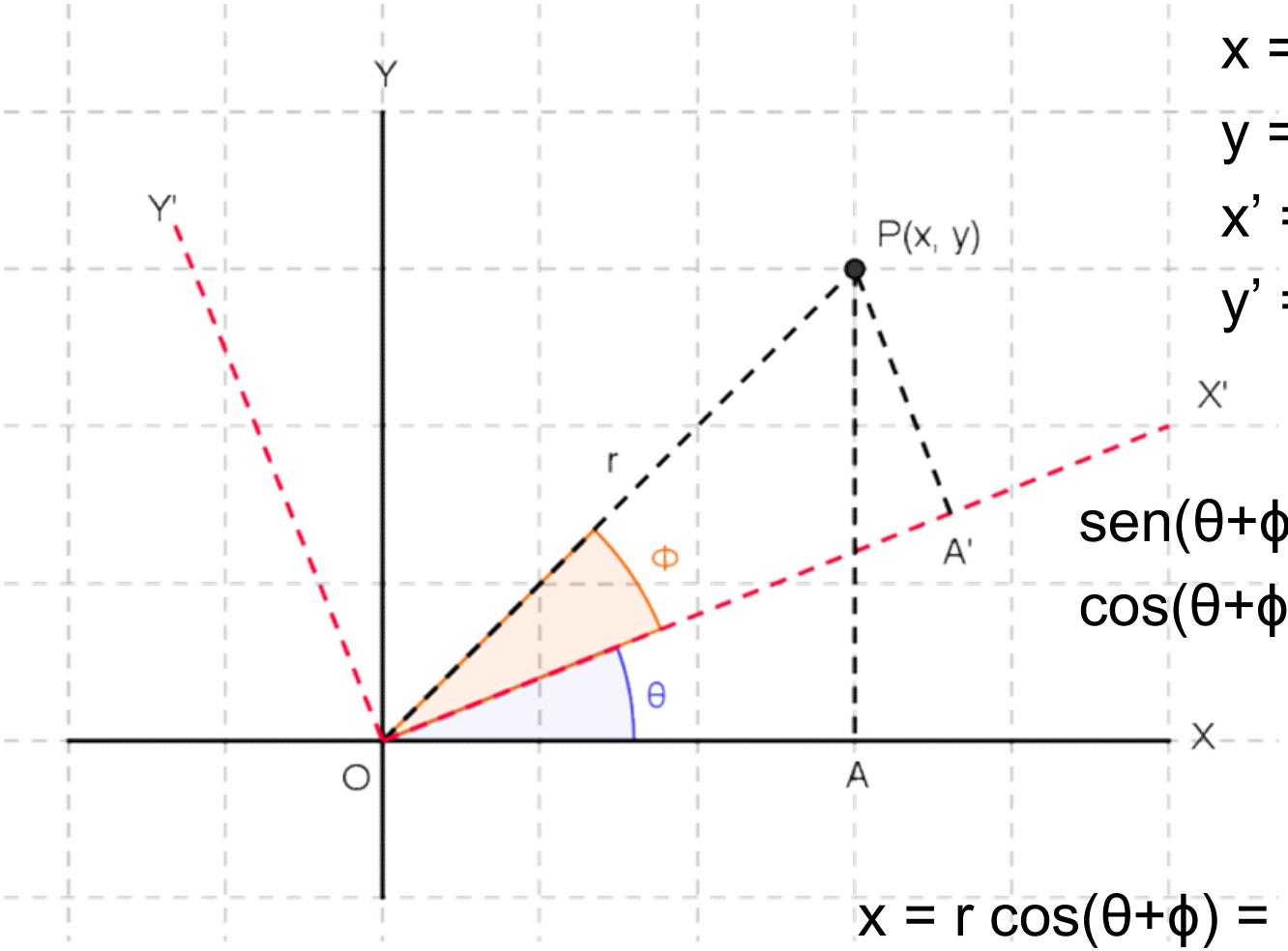
- Diferentes técnicas de generación de objetos
  - Escaneo, barrido, geometría constructiva de sólidos…
- Los objetos pueden sufrir transformaciones geométricas
  - Traslaciones
  - Escalados
  - Rotaciones
  - Deslizamientos
  - Reflejos
  - …
- Es más eficiente combinar transformaciones en una sola
  - Álgebra matricial

Rotaciones

- Para rotar una figura en el plano hay 2 alternativas que dan el mismo resultado
  - Rotar los ejes con las coordenadas fijas
  - Rotar las coordenadas con los ejes fijos
- Podemos tomar cualquier punto arbitrario como referencia
  - Para nosotros, el origen de coordenadas
- Podemos rotar con respecto a cualquier eje o par de ejes
- Ejemplo: rotando un punto (x, y) con respecto al eje Z se obtienen unas coordenadas (x', y')



Rotación sobre el eje Z



x = |OA| sen(θ) = y' / r

y = |AP| cos(θ) = x' / r

x' = |OA'| sen(θ+ϕ) = y / r

y' = |A'P| cos(θ+ϕ) = x / r

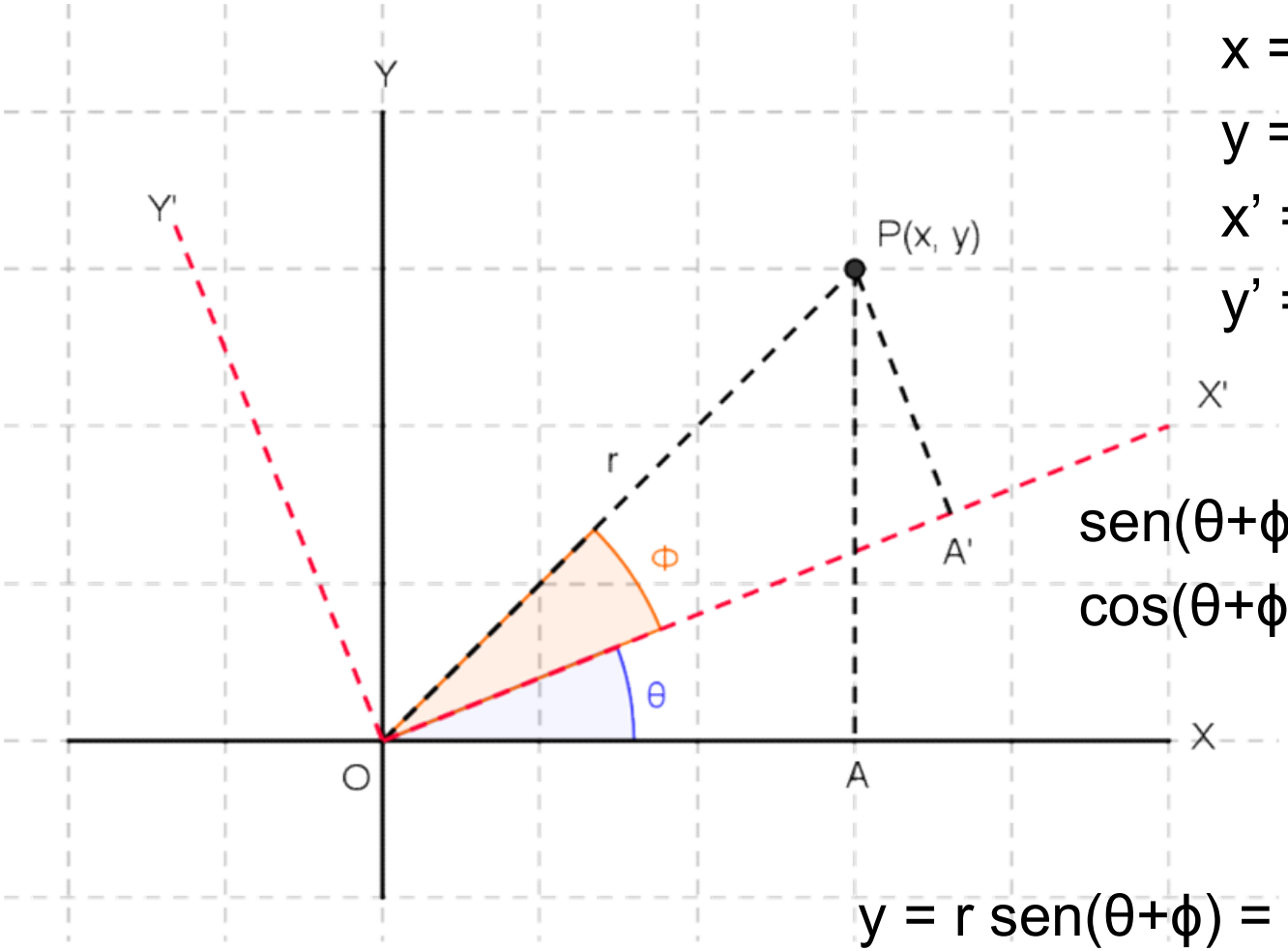
sen(θ+ϕ) = sen(θ) cos(ϕ) + sen(ϕ) cos(θ)

cos(θ+ϕ) = cos(θ) cos(ϕ) - sen(ϕ) sen(θ)

x = r cos(θ+ϕ) = r cos(θ) cos(ϕ) - r sen(θ) sen(ϕ)

x = x' cos(θ) - y' sen(θ)

Rotación sobre el eje Z



x = |OA| sen(θ) = y' / r

y = |AP| cos(θ) = x' / r

x' = |OA'| sen(θ+ϕ) = y / r

y' = |A'P| cos(θ+ϕ) = x / r

sen(θ+ϕ) = sen(θ) cos(ϕ) + sen(ϕ) cos(θ)

cos(θ+ϕ) = cos(θ) cos(ϕ) - sen(ϕ) sen(θ)

y = r sen(θ+ϕ) = r sen(θ) cos(ϕ) + r sen(ϕ) cos(θ)

y = x' sen(θ) + y' cos(θ)

Rotación sobre el eje Z

- Ecuaciones (tras un cambio de variables)
  - x' = x cos(θ) - y sen(θ)
  - y’ = x sen(θ) + y cos(θ)
- Formato matricial:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- En general, en el espacio 3D la matriz de transformación es:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Traslación

- Ecuaciones en 3D:
  - x’ = x + Δx
  - y’ = y + Δy
  - z’ = z + Δz
- Formato matricial:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x/z \\ 0 & 1 & \Delta y/z \\ 0 & 0 & 1+\Delta z/z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- La matriz de traslación depende del propio punto
- Solución: usar coordenadas homogéneas

Coordenadas homogéneas

- Coordenadas homogéneas: sistema de coordenadas que permite describir un punto en el espacio proyectivo
  - Las cartesianas, para describir puntos en geometría euclídea
  - Permiten pasar de n dimensiones a n+1
- Permiten representar las coordenadas de puntos (incluso si están en el infinito) usando coordenadas finitas
- Paso de coordenadas 3D a homogéneas:

$$(x, y, z) \rightarrow (x\omega, y\omega, z\omega, \omega)$$

con $\omega \neq 0$, típicamente $\omega = 1$

- Ahora las matrices de transformación son 4x4
  - Permite tener una matriz de traslación adecuada

Matrices en coordenadas homogéneas

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación eje X

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación eje Y

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotación eje Z

$$\begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Traslación

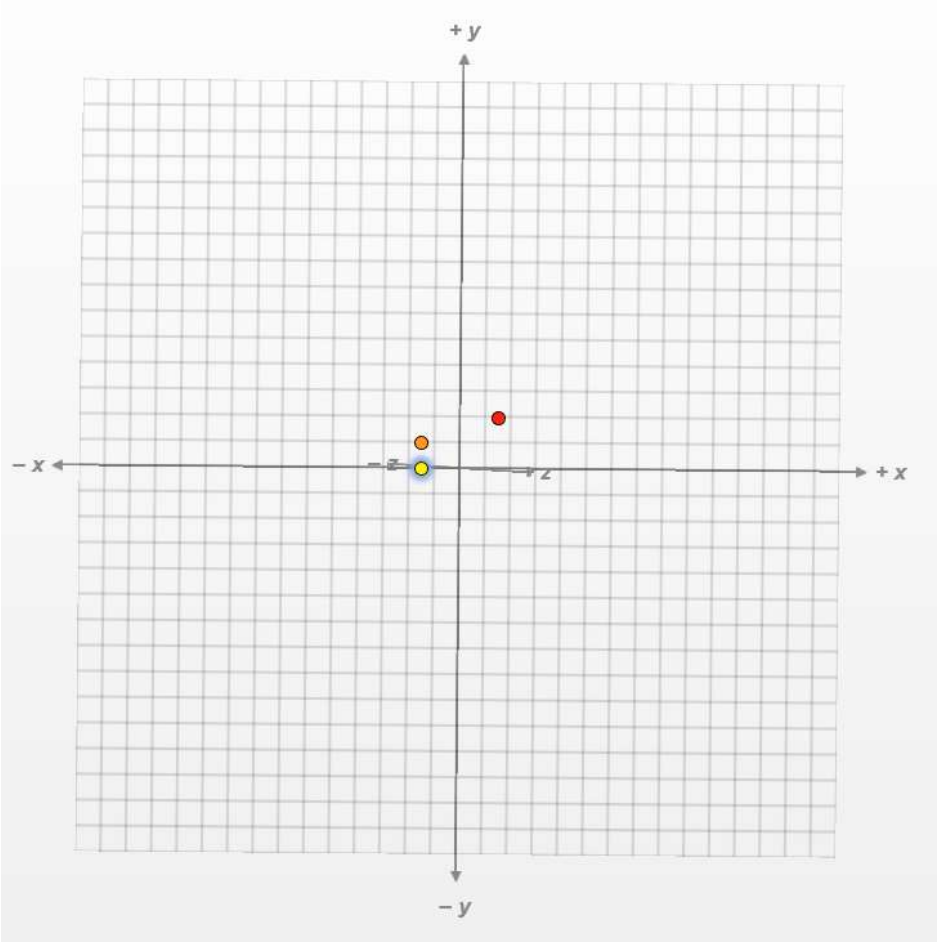
$$\begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & \mathrm{Sz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Escalado

- Composición de transformaciones: producto de matrices

Ejercicio: composición de transformaciones

- Calcular el resultado de rotar 90º el punto [1 2 3] alrededor del eje Z y después aplicarle una translación -1 en el eje Y



Ejercicio: composición de transformaciones

- Calcular el resultado de rotar 90º el punto [1 2 3] alrededor del eje Z y después aplicarle una translación -1 en el eje Y

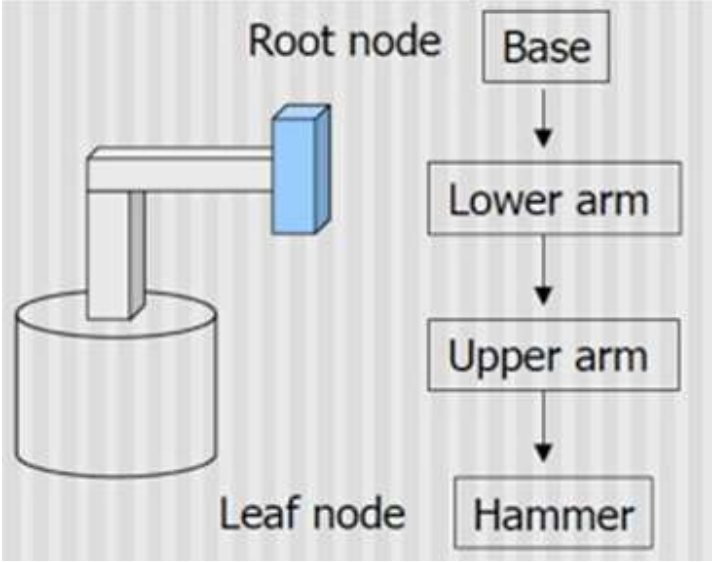
$$\begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 3 \\ 1 \end{bmatrix}$$

Modelado en estructuras jerárquicas

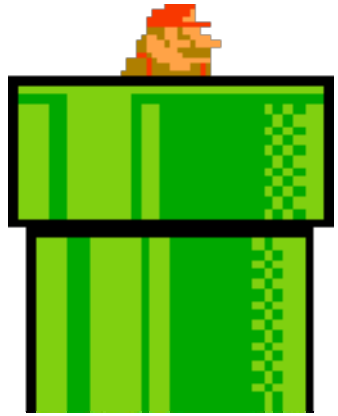
- Los modelos suelen tener estructura jerárquica de dependencias
  - Abuelos → padres → hijos
  - Mover el brazo → mover la mano → mover los dedos
- Movimientos de un nivel n se trasmiten a elementos de nivel n-1
  - Se implementa mediante herencia de transformaciones



Visualización de objetos 3D

Proceso de visualización de la imagen

- Tubería de renderizado (*rendering pipeline*): transformación de la representación de una escena 3D en una imagen 2D para la pantalla de un ordenador u otro dispositivo hardware
- Pasos:
  1. Modelado (en sistema de coordenadas 3D global)
  2. Transformación de vista
  3. Recorte frontal y trasero
  4. Proyección
  5. Visualización (paso a pantalla del ordenador)



Modelo de cámara

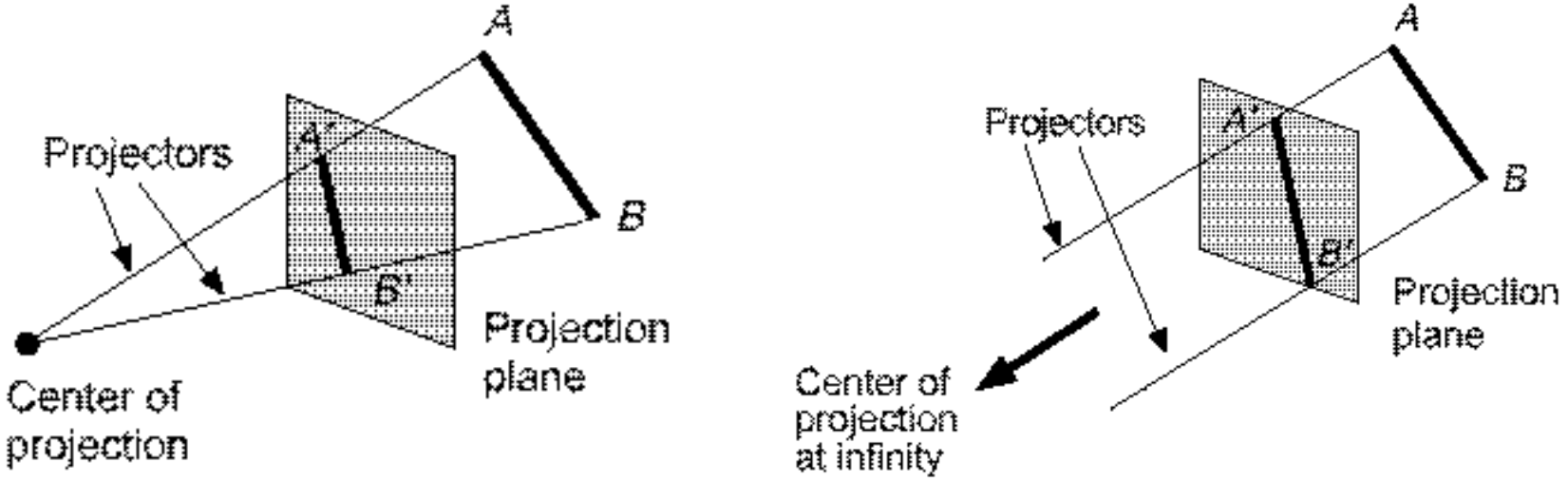
- Los objetos son, básicamente, posiciones en el espacio (geometría), junto con relaciones (topología), que son interpretadas según las capacidades gráficas, dependiendo de las necesidades del usuario
- Típicamente, se forman estructuras jerárquicas agrupando objetos en niveles
- ¿Qué hace el ojo o una cámara?
  - Selecciona una parte de la escena, eliminando el resto
  - Proyecta la información en una superficie bidimensional (la retina o la película)
- Procesos que vamos a distinguir:
  - Posicionamiento y orientación
  - Proyección

Proyección

- Paso de un espacio de n dimensiones a otro de n−1
- Normalmente, de 3 (mundo) a 2 (medios de representación)
- Parámetros:
  - Centro de proyección: un punto en el espacio, a través del cual pasan todos los proyectores
  - Proyector: recta formada entre el centro de proyección y cada uno de los puntos que son proyectados
  - Plano de proyección: superficie de una dimensión menor a la del espacio donde se realiza la proyección
- Proyección: intersección (conjunto de vértices) de los proyectores con el plano de proyección

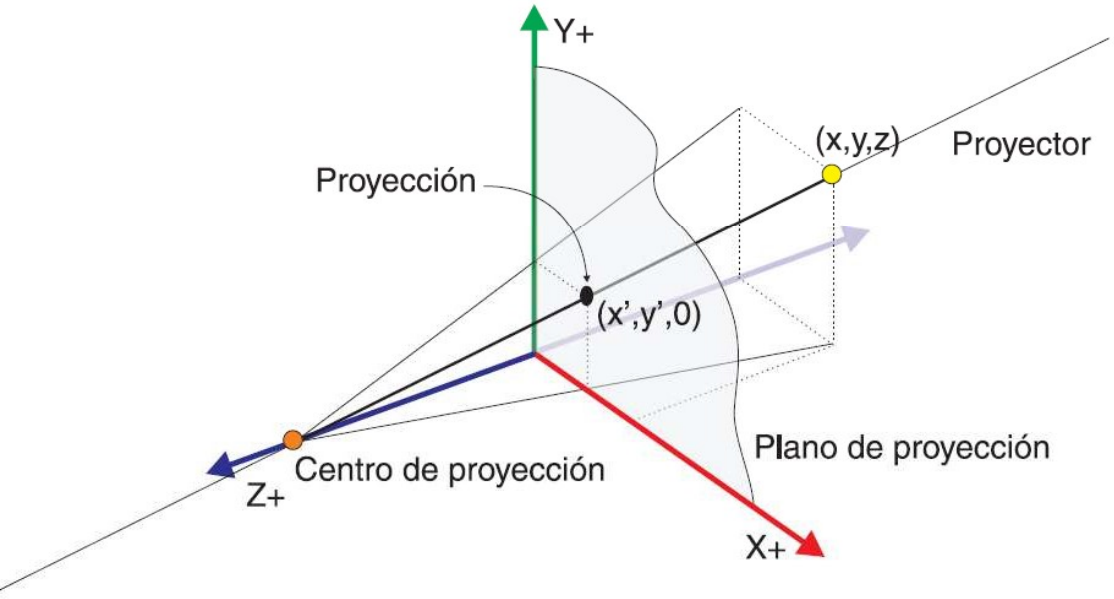
Proyección paralela y perspectiva

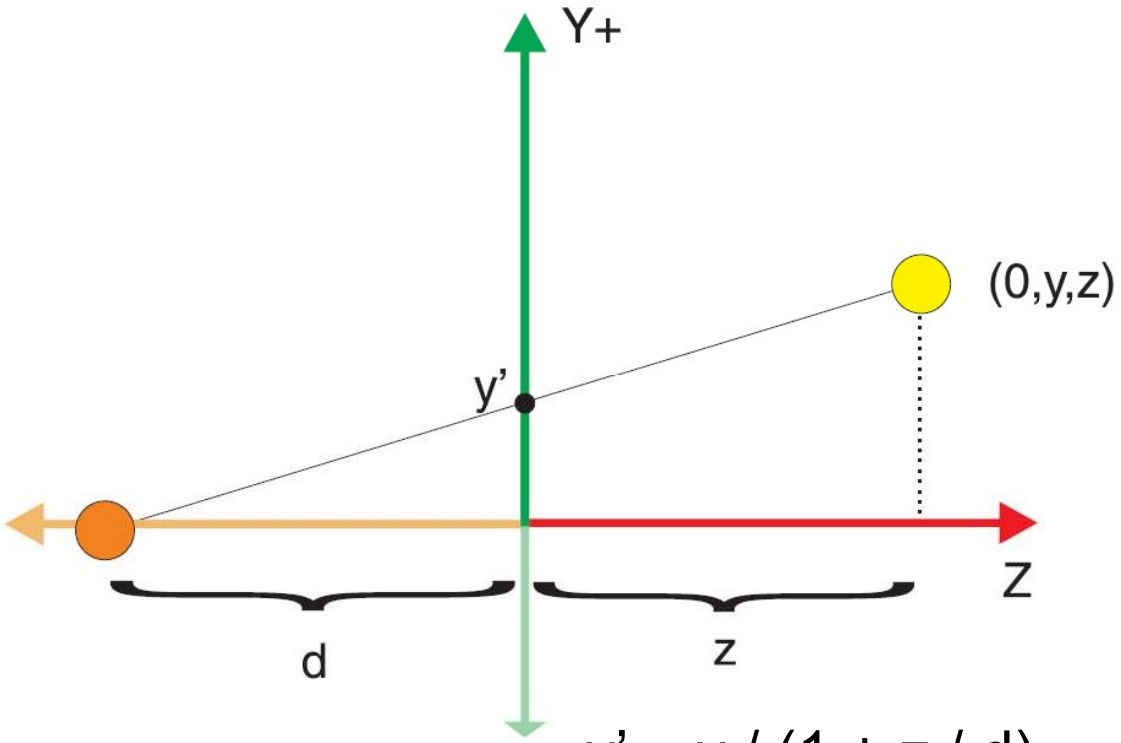
- Tipos de proyecciones
  - Paralela: centro de proyección en el infinito
  - Perspectiva: centro de proyección en una posición finita



- Punto de fuga: punto donde convergen 2 líneas paralelas
  - En la proyección perspectiva, habrá x puntos de fuga si el plano de proyección corta x ejes, $x \in \{1, 2, 3\}$

Proyección de perspectiva





Suponemos que el plano de proyección es el eje Z

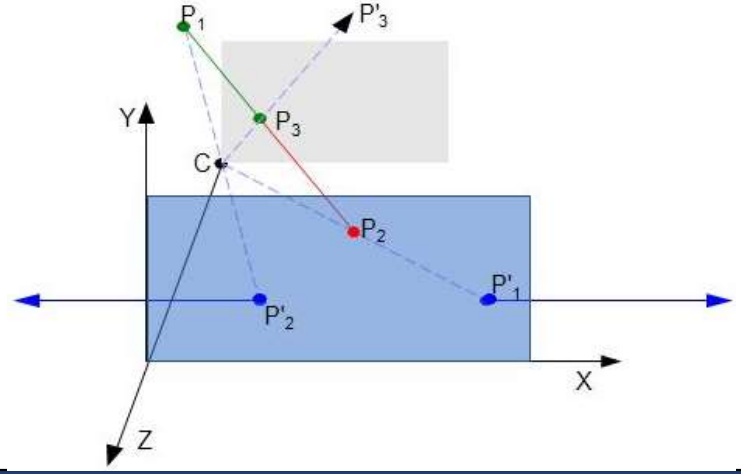
$$y' = y / (1 + z / d)$$
$$x' = x / (1 + z / d)$$

Matriz de transformación

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/-d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Proyección de perspectiva: propiedades

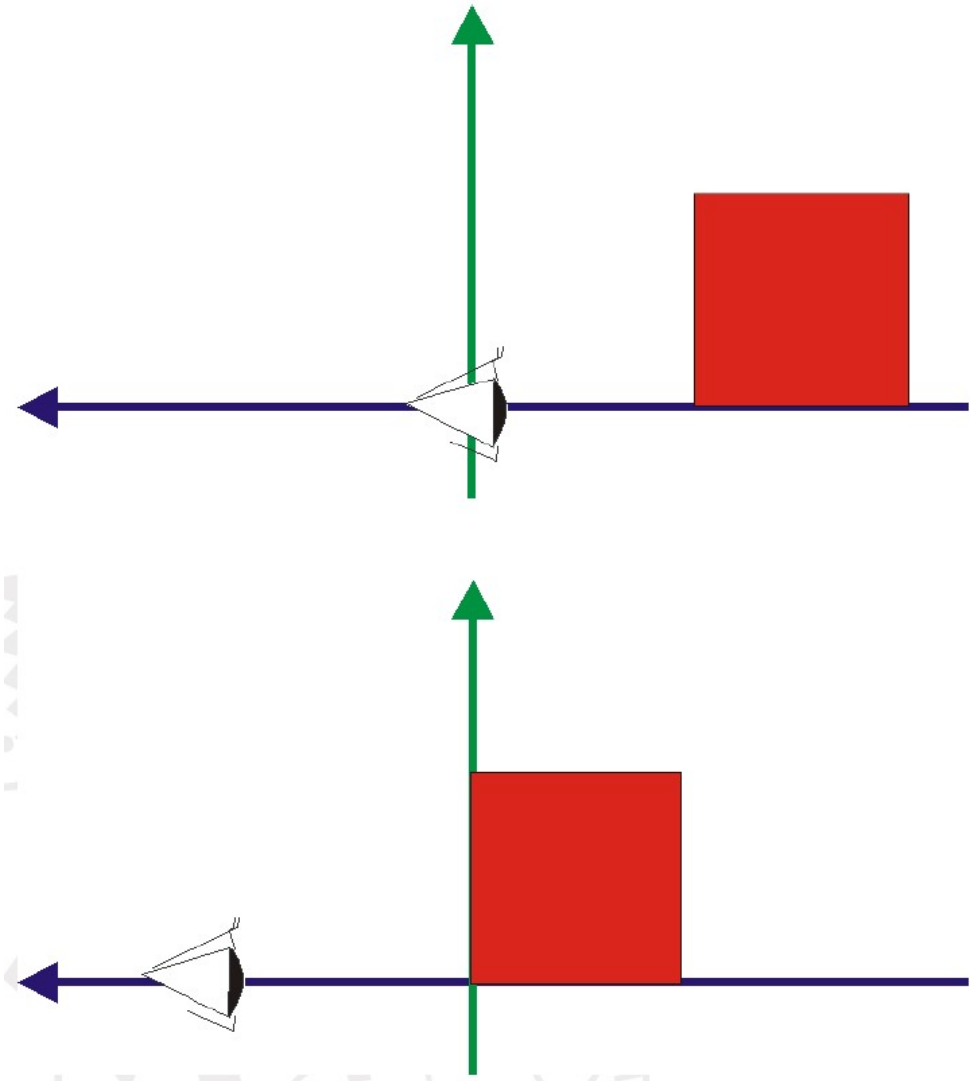
- Acortamiento perspectivo: los objetos se ven más pequeños según aumenta su distancia al centro de proyección
- Puntos de fuga: varias rectas paralelas parecen confluir en un mismo punto (como las vías de ferrocarril)
- Confusión de vista: la orientación de los objetos cambia dependiendo si están delante o detrás del plano
- Distorsión topológica: segmentos finitos dan rectas discontinuas infinitas si hay partes delante y detrás del centro de proyección



Transformación de vista

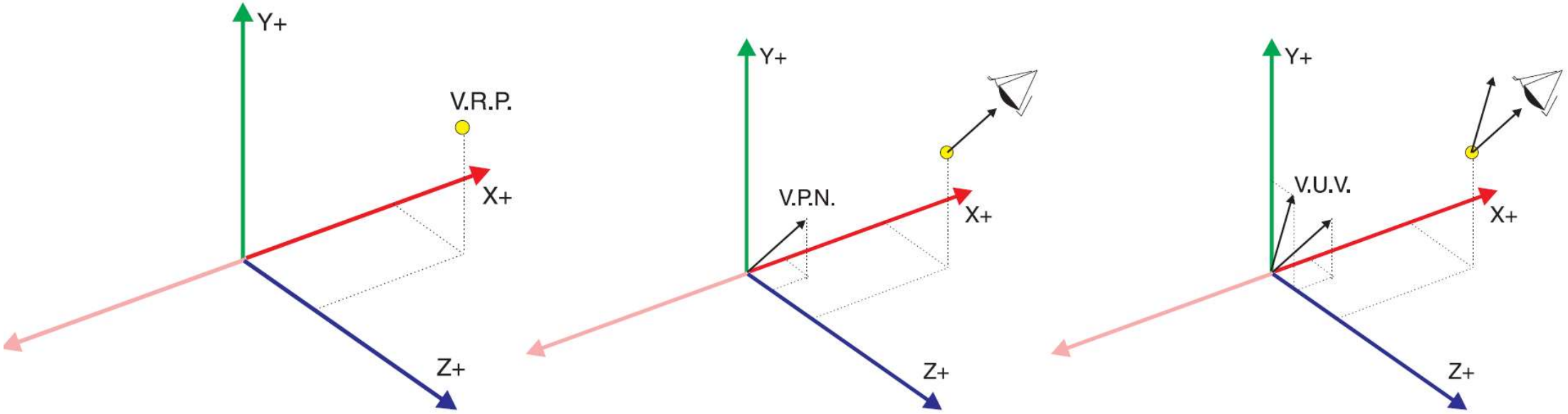
- Es necesario poder observar la escena desde cualquier posición; la cámara no tiene por qué estar en el origen
- Para ello, veremos la colocación y orientación de la cámara
- Idea básica: convertir los movimientos de la cámara virtual en los movimientos inversos u opuestos en los objetos
- Será necesario definir un sistema de coordenadas de vista diferente al sistema de coordenadas del mundo
- Para definir el posicionamiento de la cámara:
  - Definición de los parámetros de la cámara
  - Construcción del Sistema de Coordenadas de Vista
  - Cálculo de la Transformación de Vista

Transformación de vista



Parámetros de la cámara

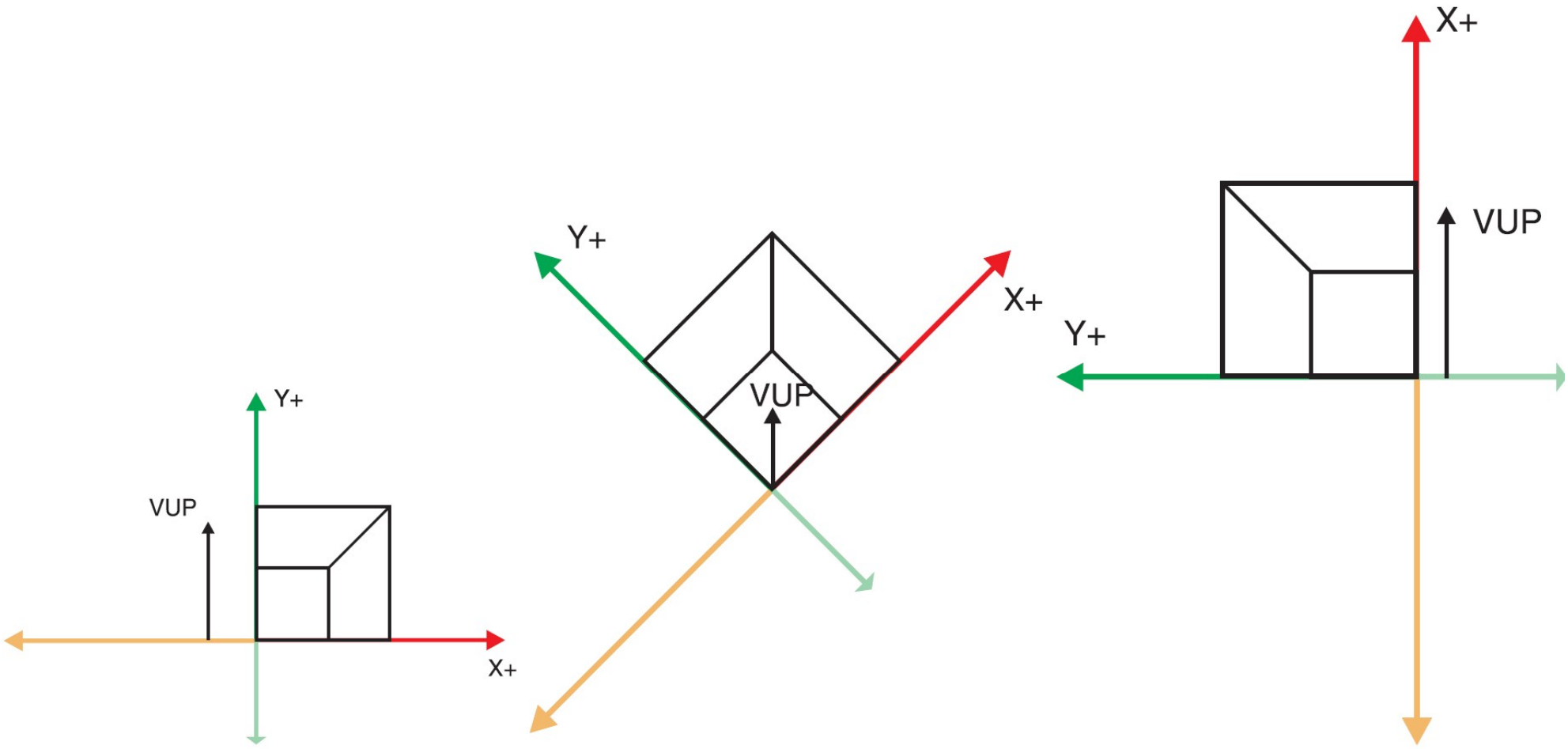
- Posición de la cámara (VRP, *view reference point*)
- Orientación hacia donde se mira (VPN, *view plane normal*)
- Verticalidad "hacia arriba" (VUP, *view up;* VUV, *view up vector*)



Parámetros de la cámara

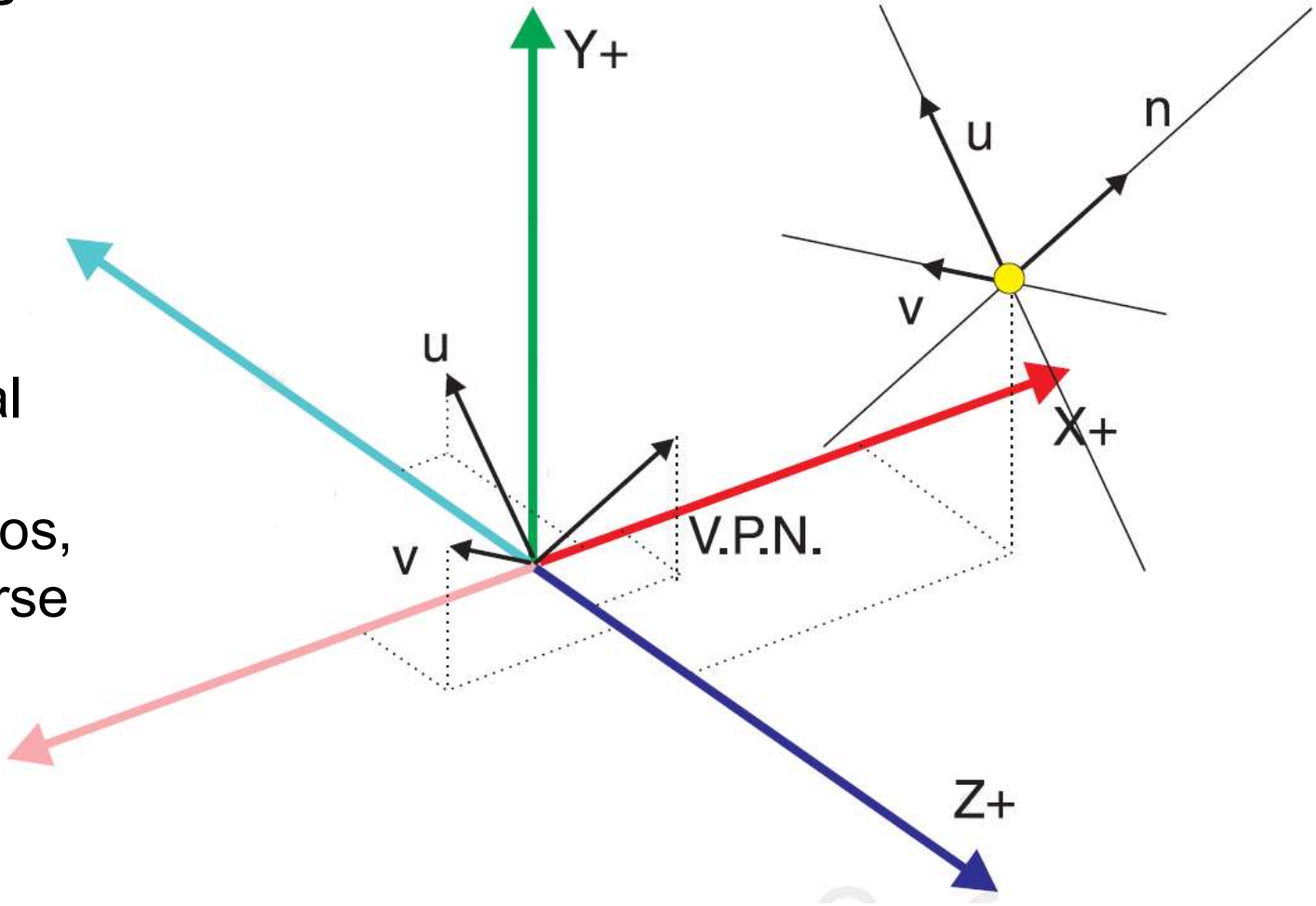
- Posición de la cámara (VRP, *view reference point*)
  - Se indica mediante un punto dado con respecto al sistema de coordenadas fijo (de mundo) (x, y, z)
  - Por este punto pasa el plano de proyección
- Orientación hacia donde se mira (VPN, *view plane normal*)
  - Vector con respecto al sistema de coordenadas fijo
  - El punto del vector es el origen
  - El sentido del vector es opuesto hacia donde se mira
  - El vector es la normal (perpendicular) al plano
- Verticalidad "hacia arriba" (VUP, *view up;* VUV, *view up vector*)
  - Vector con respecto al sistema de coordenadas fijo
  - No puede ser paralelo al VPN

Efecto del VUP



Sistema de coordenadas de vista

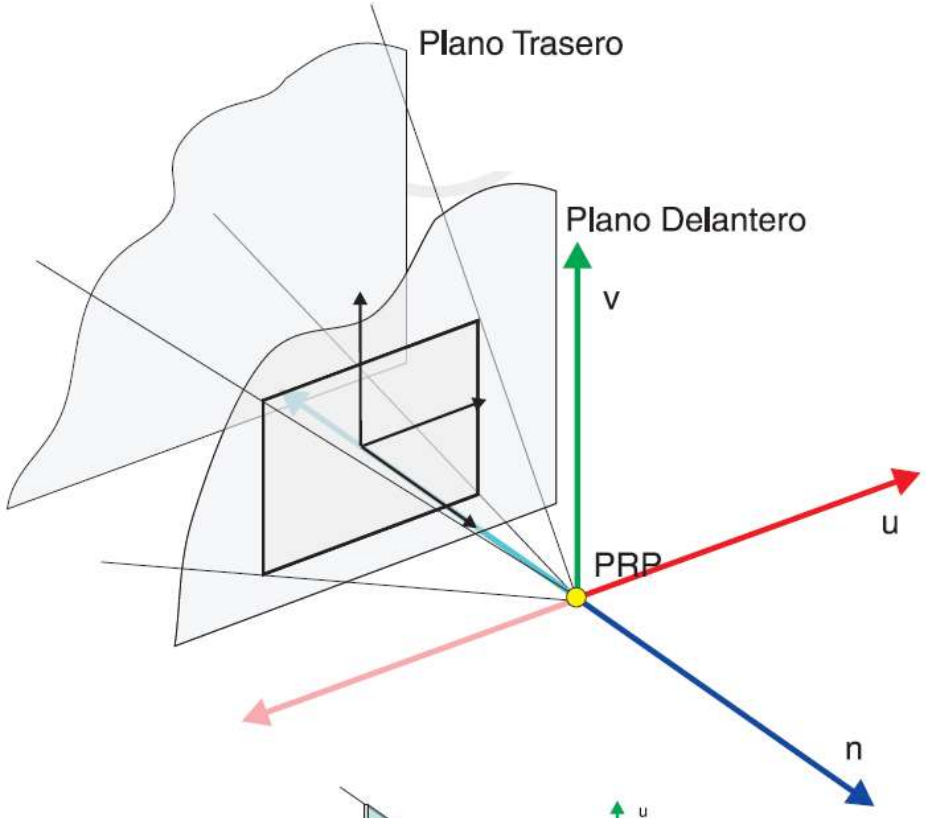
1. Origen del sistema: VRP
2. Eje Z (llamado n): VPN
3. Eje X: u = VUP ⊗ VPN
   - ⊗: producto vectorial
   - Si fueran paralelos, ⊗ no podría calcularse
4. Eje Y: v = VPN ⊗ u

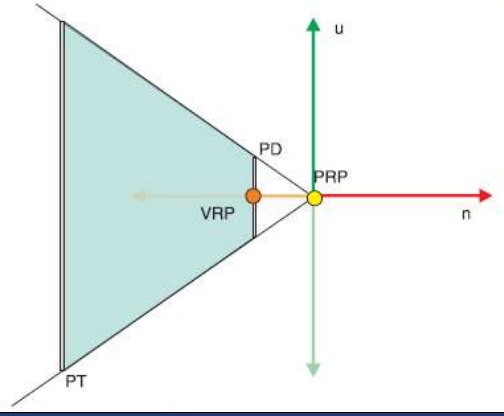


Cálculo de la transformación de vista

- Transformación de vista:
  - Conjunto de transformaciones geométricas que permiten hacer coincidir el SCV con el SCM
- Algoritmo:
  1. Traslación de la cámara (VRP) al origen del Sistema de Coordenadas del Mundo
  2. Rotación con respecto al eje X
  3. Rotación con respecto al eje Y
  4. Rotación con respecto al eje Z

! Las rotaciones, en orden arbitrario





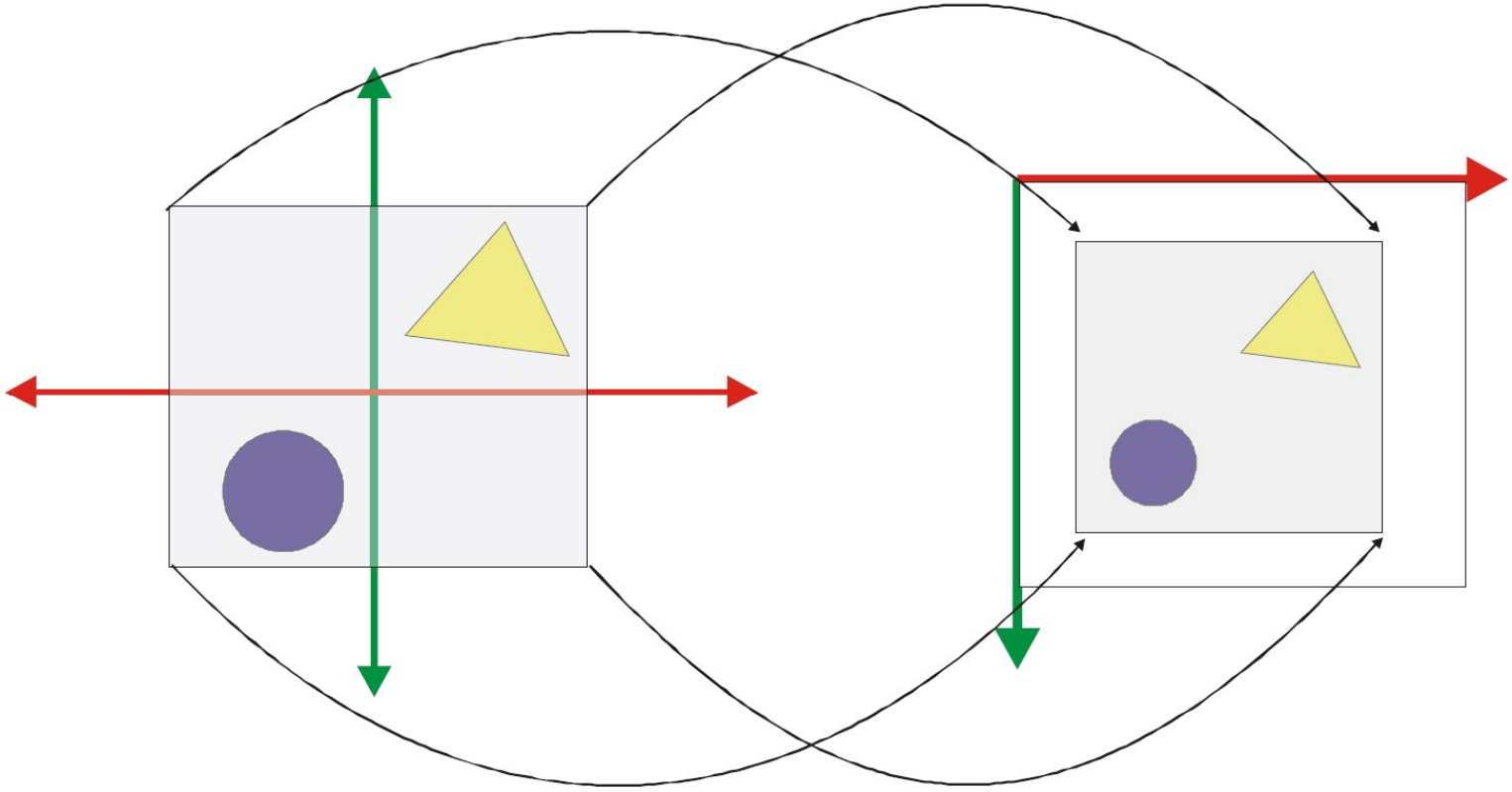
Visualización de objetos 2D

Sistemas de Coordenadas de Dispositivo

- Imagen: conjunto de píxeles dispuestos de forma matricial
- Píxel: elemento pictórico básico
- Para acceder a cada píxel hay que indicar una posición, generalmente en el sistema de coordenadas de dispositivo
  - Sistema de coordenadas bidimensional (fila, columna)
  - Coordenadas enteras (no podemos dibujar medio píxel)
  - Coordenadas positivas
  - Normalmente el origen es la esquina superior izquierda
  - Rango limitado (capacidad de dispositivos, memoria…)
- Para evitar las limitaciones, se trabaja en el sistema de coordenadas de mundo y se transforma antes de dibujar

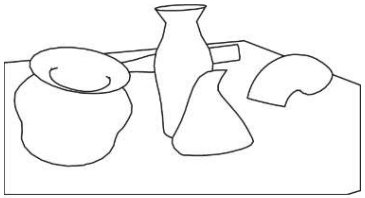
Transformación de visualización

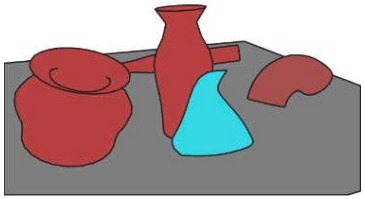
- Transformación de visualización
  - Permite pasar de sistema de coordenadas de mundo a sistema de coordenadas de dispositivo



Pixelación o renderización

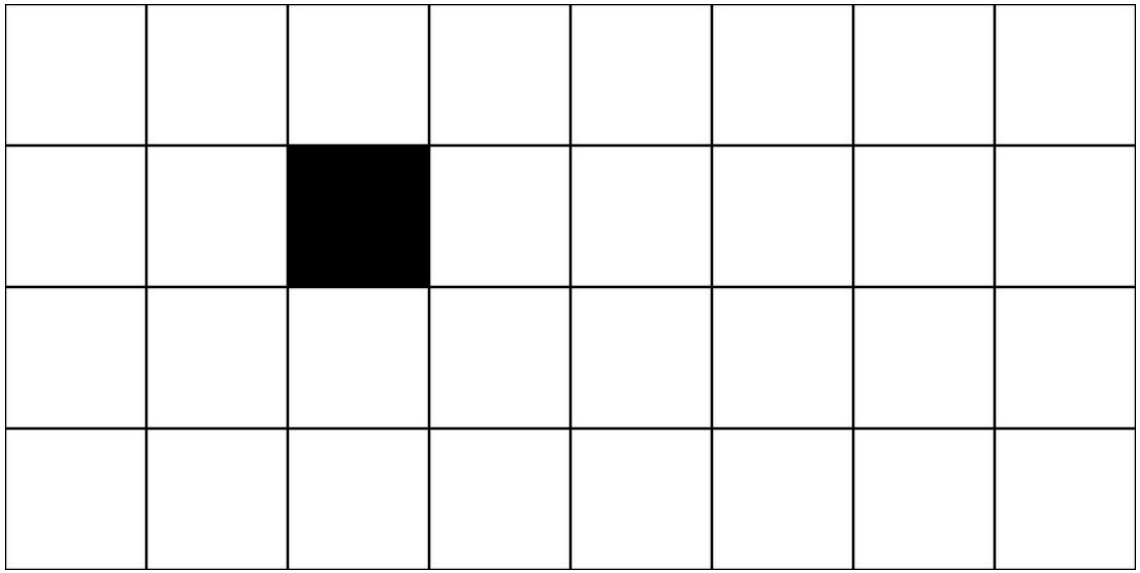
- Pixelación, renderización o rasterización: proceso que permite convertir las primitivas gráficas (descripción matemática de elementos gráficos) en el conjunto de píxeles que los representan
- Las imágenes son aproximaciones
  - Mejores cuanto mayor sea la resolución
- Primitivas gráficas para la representación en coordenadas de dispositivo de puntos, líneas, círculos, polígonos sencillos…
- Tipos de posicionado
  - Absoluto: en coordenadas del dispositivo
  - Relativo: en relación a otros elementos





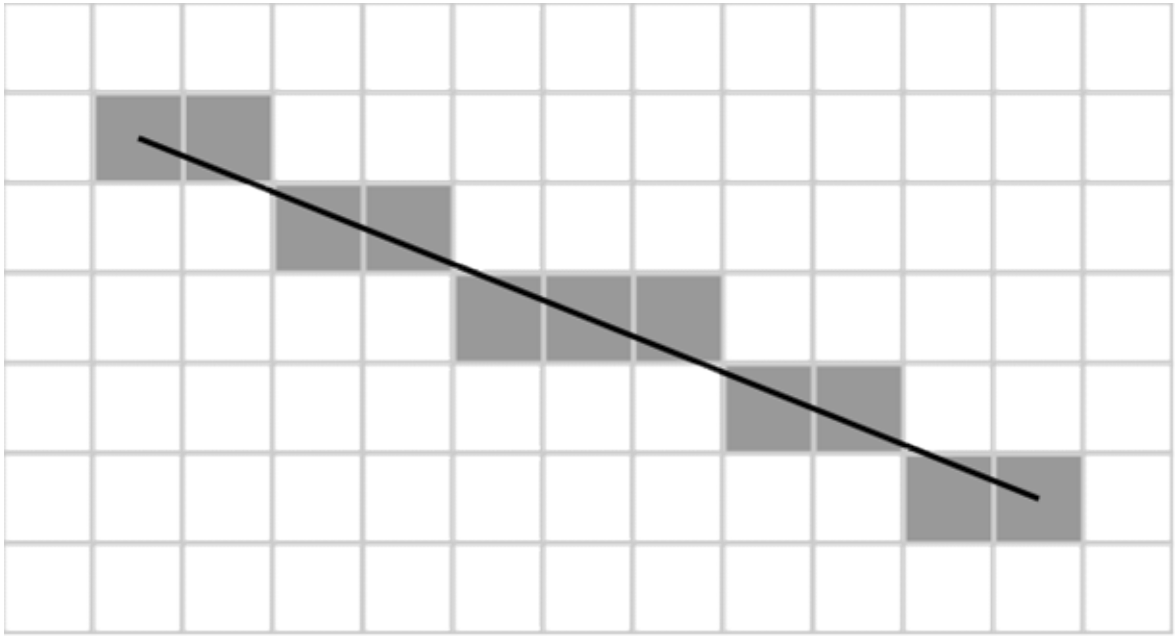
Pixelación de un punto

- Pixelación de un punto
  - Un punto se convierte en un área
  - La clave es qué posición dentro del área es la referencia
  - Las demás primitivas se construyen a partir de esta
  - Ejemplo: punto en la posición (3,2)



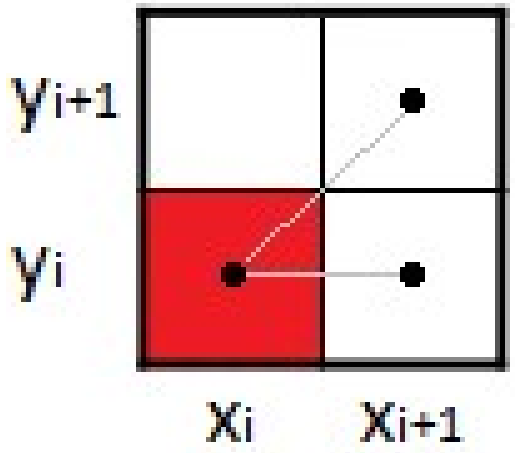
Pixelación de una recta

- Proceso:
  - Obtener los píxeles inicial y final
  - Obtener los píxeles intermedios
- Muy fácil para ciertos tipos de líneas rectas
  - Horizontales
  - Verticales
  - Con pendiente ± 1
- Ej: recta de (2,2) a (12,6)



Pixelación de una recta

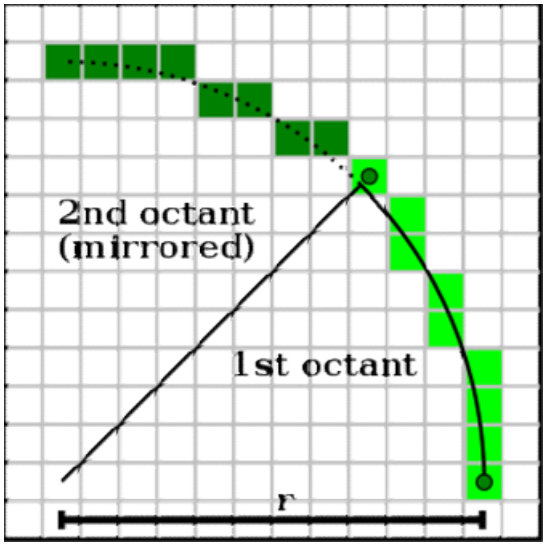
- Algoritmo de Bresenham
  - Tras dibujar el píxel (x_i, y_i) y si la pendiente $m \in (0,1)$, hay solo 2 opciones para el siguiente: (x_{i+1}, y_i) o (x_{i+1}, y_{i+1})



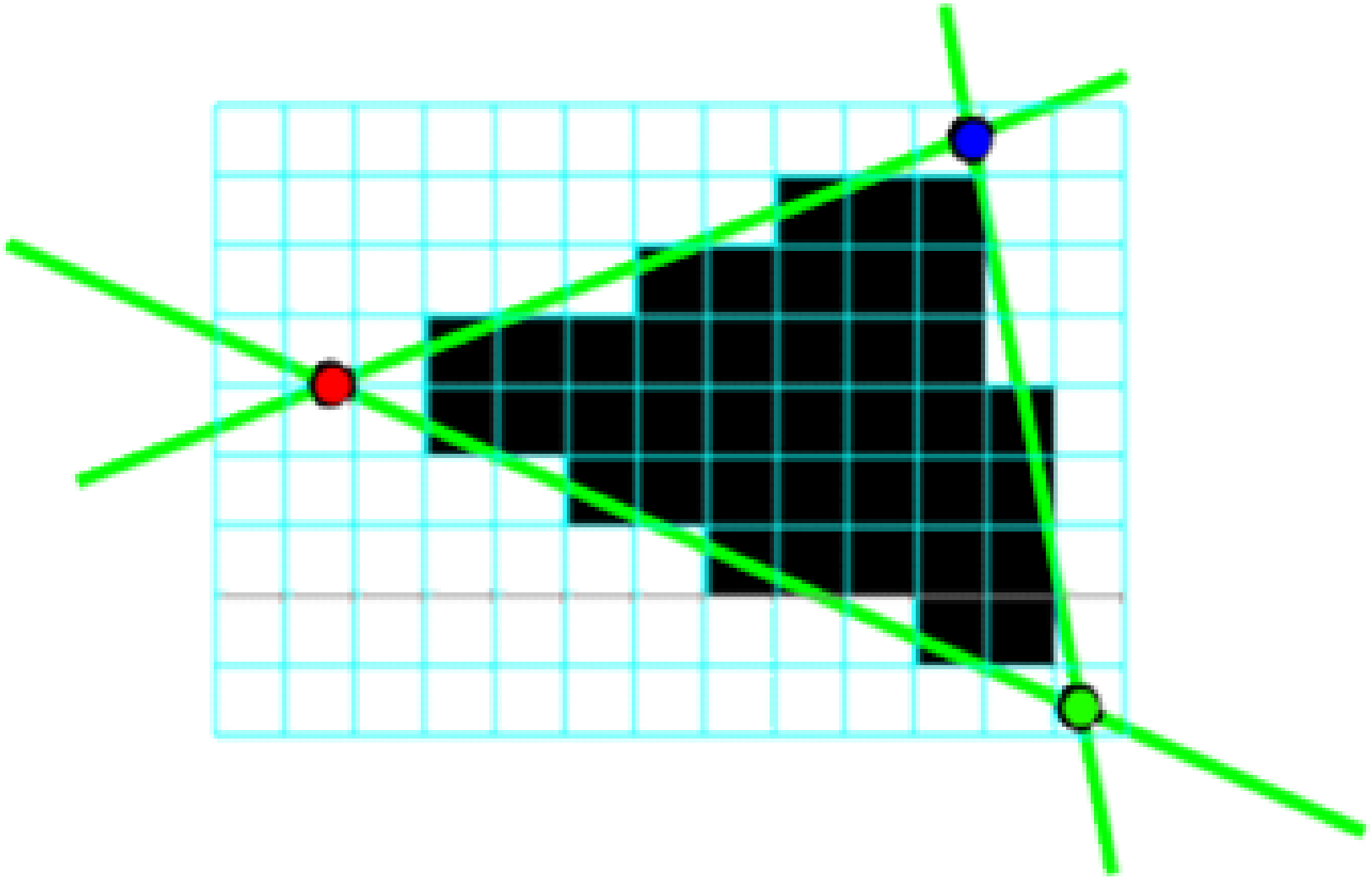
  - Seleccionar aquél que se aproxime más a la recta real
  - Es un algoritmo de error incremental

Pixelación de un círculo

- Se puede extender el algoritmo de Bresenham
- Resolver por octantes de 45º y usar simetría
- La pendiente es variable: al principio hay muchos puntos juntos, pero luego hay una gran variación
- Costoso calcular senos/cosenos (coordenadas polares)

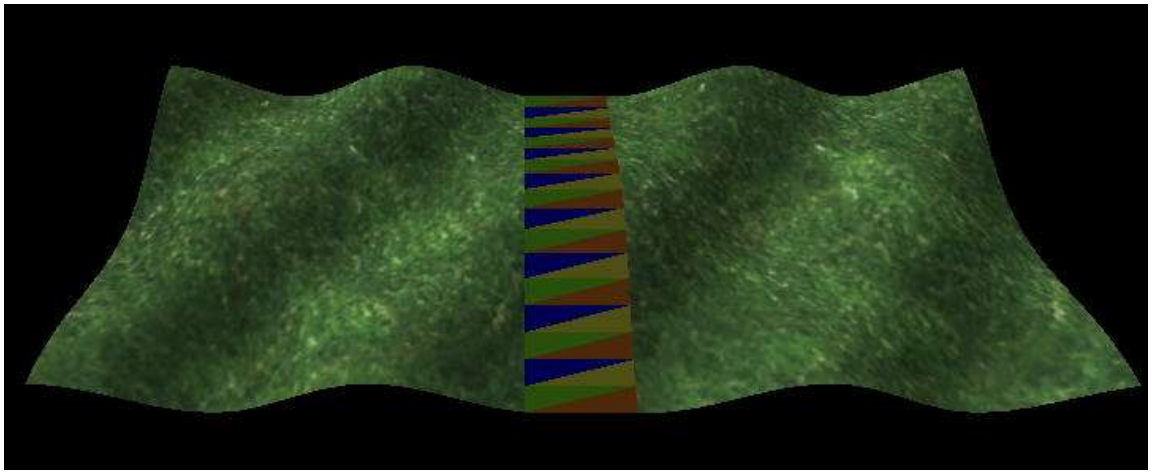


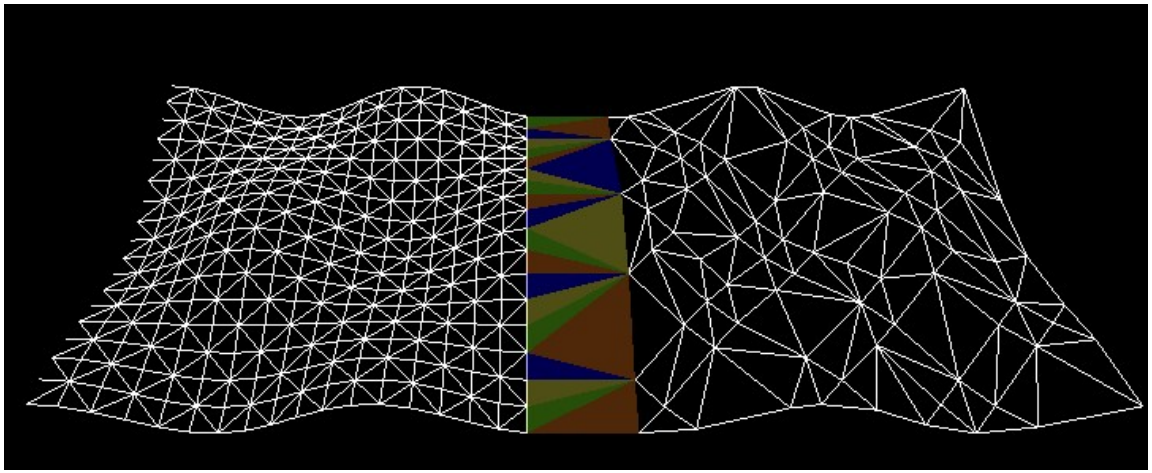
Pixelación de un polígono



Pixelación de un polígono

- El caso básico es el triángulo; los demás pueden reducirse a ese





Pixelación de un polígono

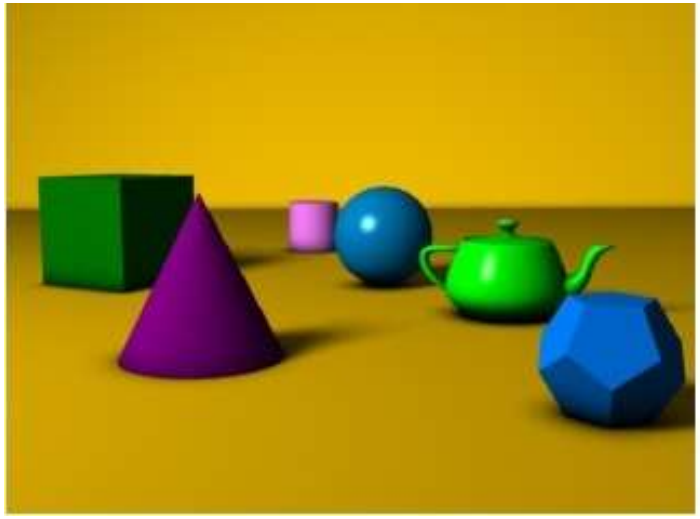
- El caso básico es el triángulo; los demás pueden reducirse a ese

Pixelación de un polígono

- El caso básico es el triángulo; los demás pueden reducirse a ese

Z-búfer

- z-búfer
  - Almacena la profundidad de cada píxel (coordenada z) para decidir qué elementos son visibles y cuáles ocultos
  - Cuando un objeto se renderiza en un píxel, se compara su profundidad con la almacenada para dicho píxel y solo se mantiene el color de la más cercana al observador
  - Se puede implementar por hardware o software



Investigación en la Universidad de Zaragoza

Modelado geométrico

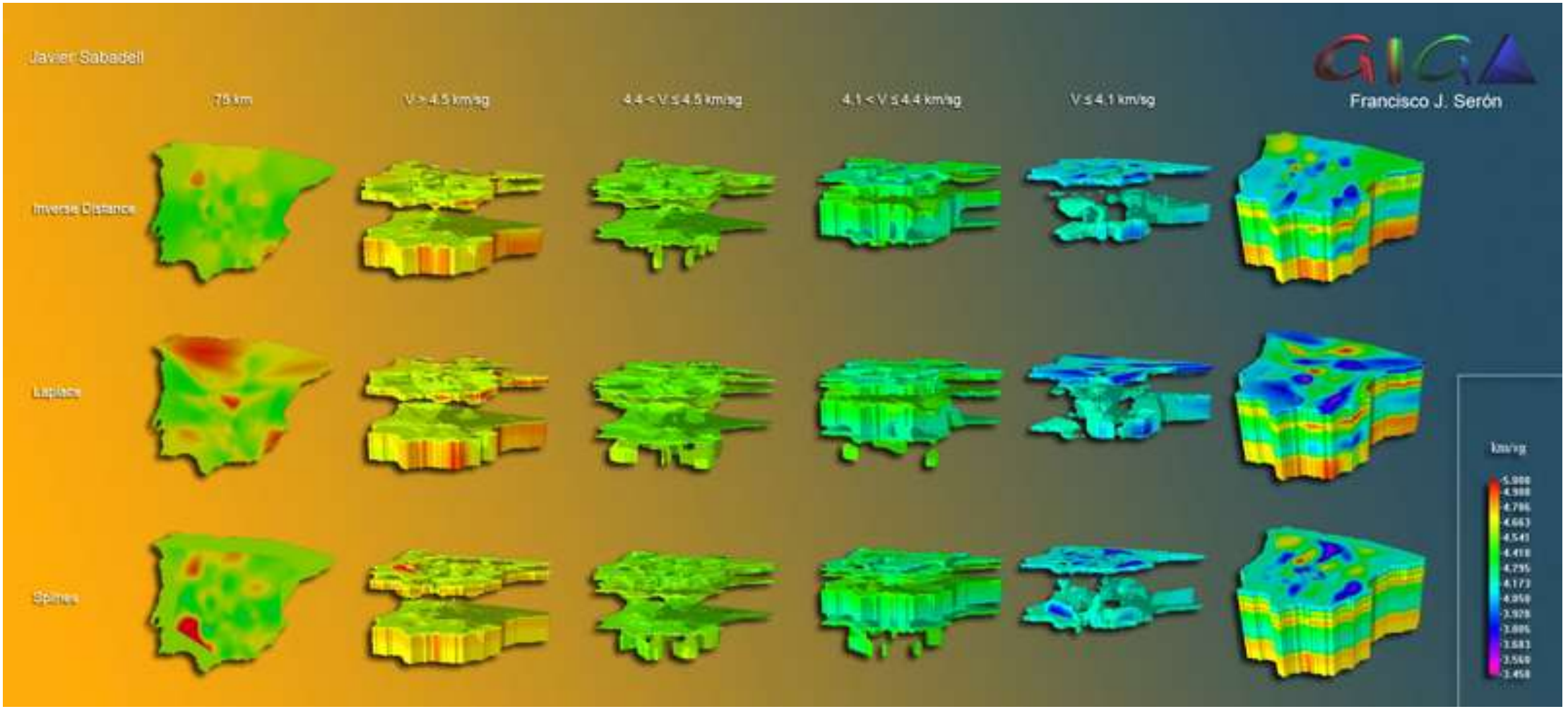
Modelado visual

Iluminación

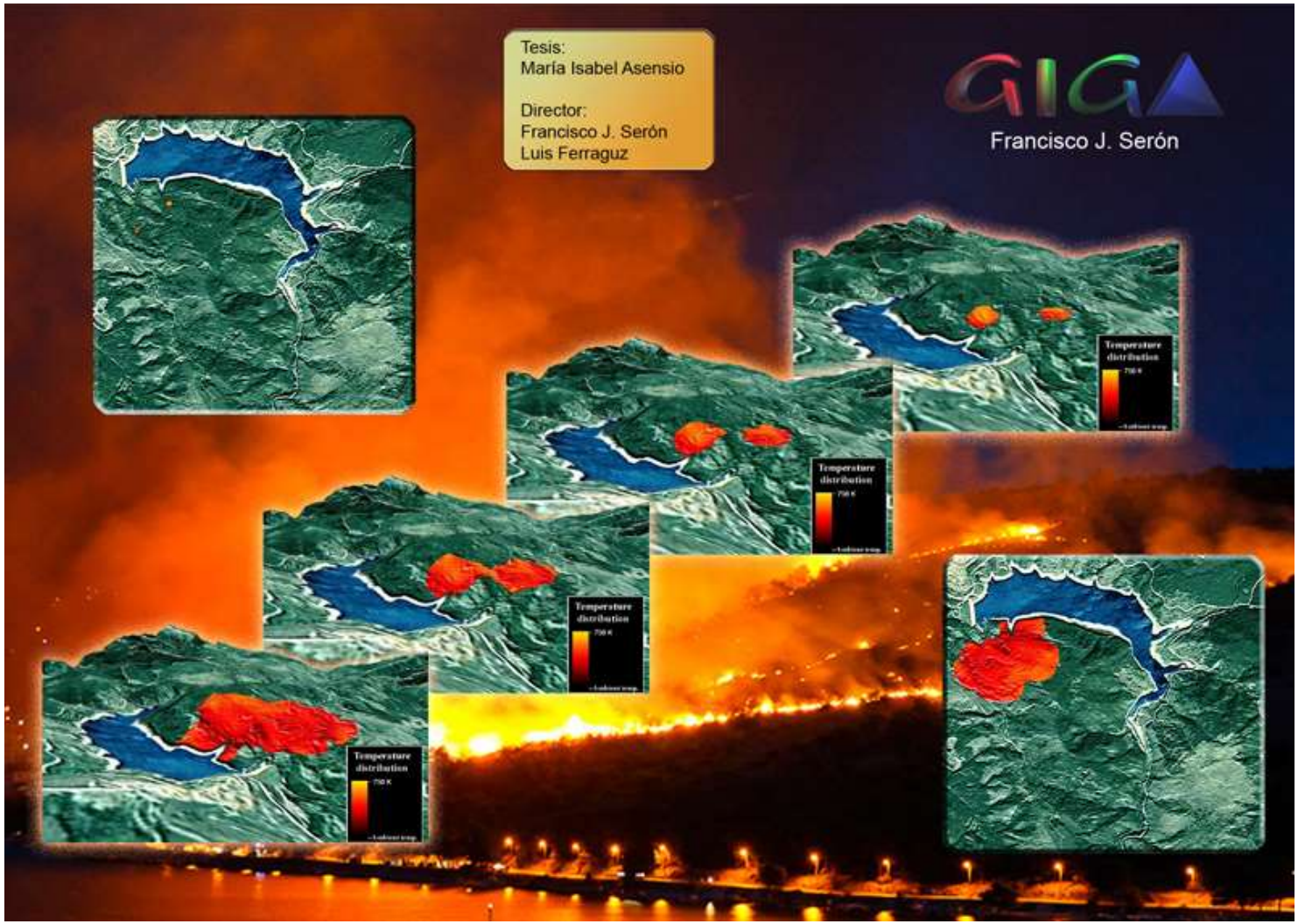
Animación


GIG
Francisco J. Serón
Tesis:
Sandra Baldassarri
Director:
Francisco J. Serón

Visualización de datos

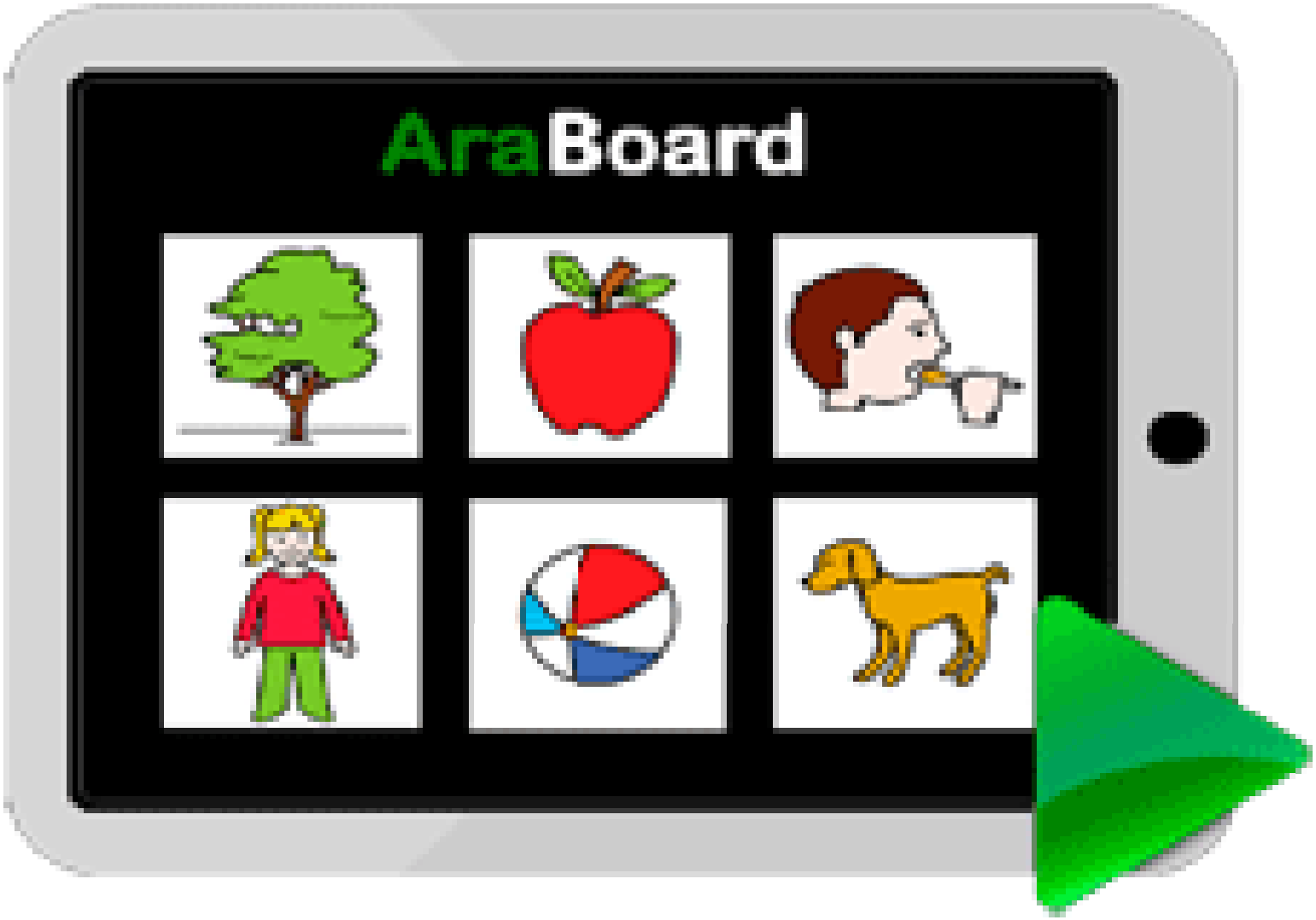


Simulación


Tesis:
María Isabel Asensio
Director:
Francisco J. Serón
Luis Ferraguz
GIGA
Francisco J. Serón

Agentes cognitivos

Accesibilidad



Videojuegos

Realidad virtual

Realidad aumentada