

Fundamentos de Computación Gráfica

Proyecciones Gráficas

Juan Carlos Arbeláez E

Universidad Eafit

Tabla de contenido

Formación de imágenes

Cámaras y lentes

Modelo Cámara Pinhole

Lentes

Thin lenses

Profundidad de campo

De 3D a imagen digital

Transformaciones de sistemas coordenados

Flujo de transformaciones

Determinación de caras visibles

Back-face elimination

Depth-Sort

Z-Buffer

Bibliografía

Proyecciones gráficas



¿Son los autos del mismo tamaño en la imagen? Tomado de brainbashers.com

Introducción

Formación de imágenes

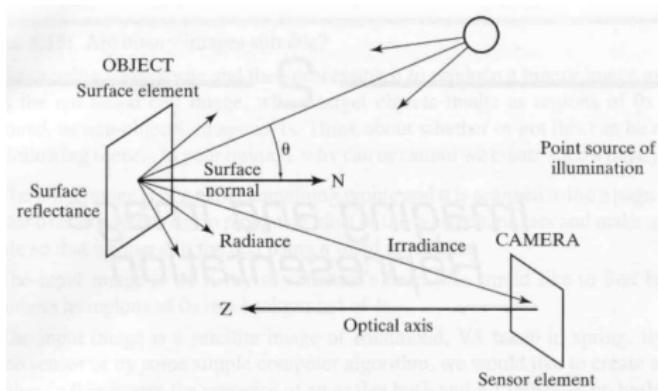
Compuesta de:

1. Geométrica

correspondencia 3D -
2D

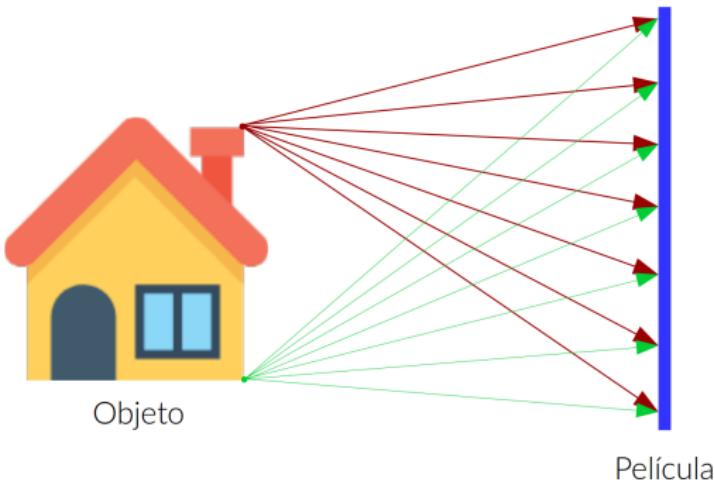
2. Física de la luz

Brillo de cada punto
de la imagen



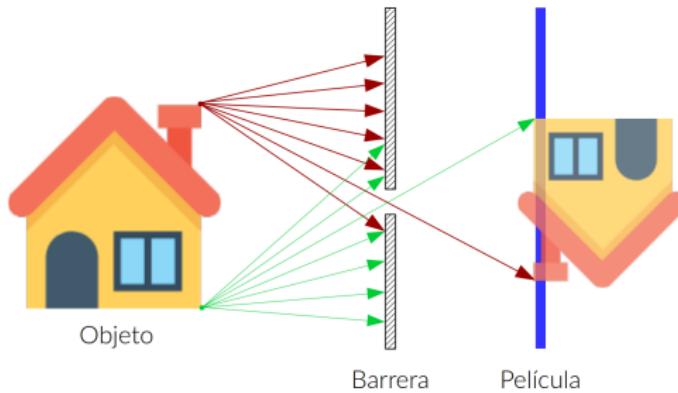
Modelo simplificado de la formación de imágenes. Tomado de Computer Vision - University of Nevada

Formación de imágenes



- ▶ ¿Si diseñamos una simple cámara?
- ▶ ¿Cómo resultaría la imagen?

Formación de imágenes

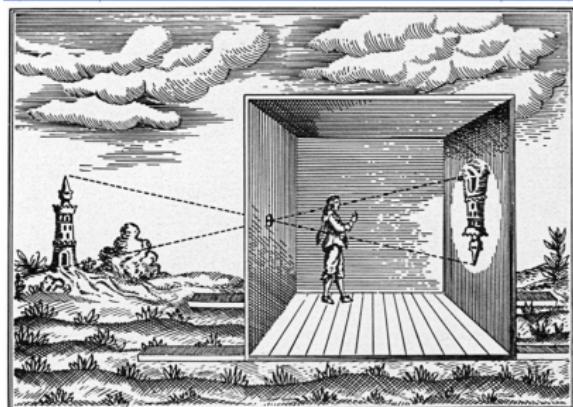


- ▶ Agregamos una barrera para bloquear mayoría de rayos
- ▶ Reducimos la borrosidad
- ▶ ¿Cómo se transforma la imagen?

Formación de imágenes

Cámara oscura

- ▶ Transferir una imagen compuesta por luz
- ▶ Pieza oscura y un agujero en una pared
- ▶ Estudio de perspectiva y eclipses
- ▶ No existían medios fotosensibles

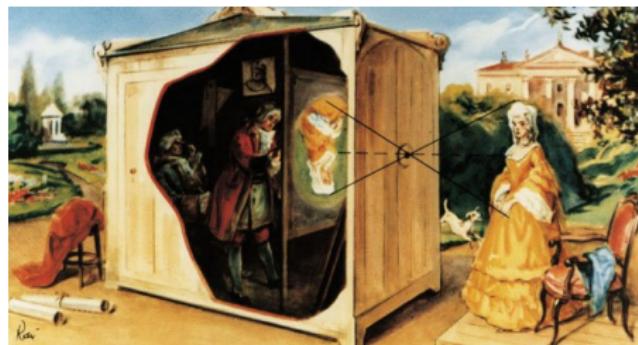


Tomado de photoion.co.uk

Cámara oscura

Usos

- ▶ Estudio de perspectiva
- ▶ Visualización de eclipses
- ▶ Ayuda para pintar

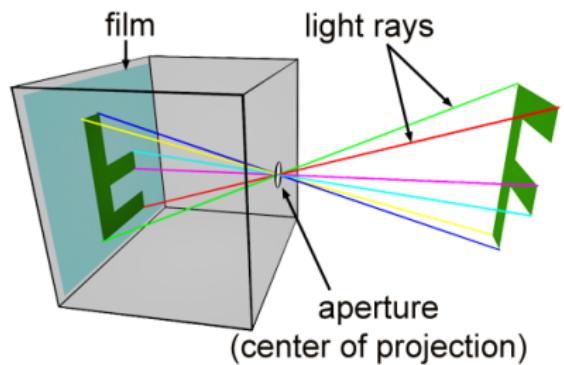


Tomado de photoion.co.uk

Formación de imágenes

Pinhole cámara

- ▶ Cámara oscura con un medio fotosensible
- ▶ Rayos pasan por el agujero
- ▶ Forma imagen rotada 180° con respecto al eje óptico



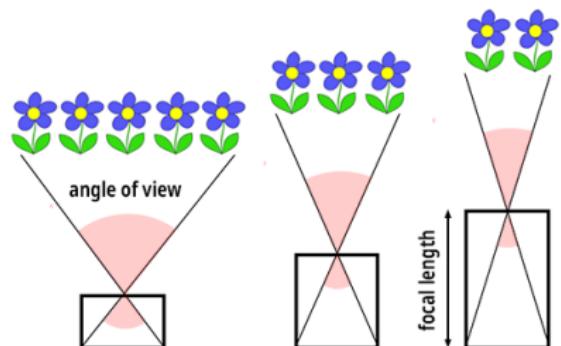
Tomado de scratchapixel.com

Parámetros de cámara *Pinhole*

Distancia focal

Distancia entre la apertura y la imagen del plano.

Define el ángulo de visión y dictamina la apertura requerida para enfocar



Tomado de scratchapixel.com

Parámetros de cámara *Pinhole*

Diámetro de la apertura

Define que tan agudo y brillante es la imagen.

Muy pequeño, muy poca luz ingresa y es sujeto a efectos de difracción



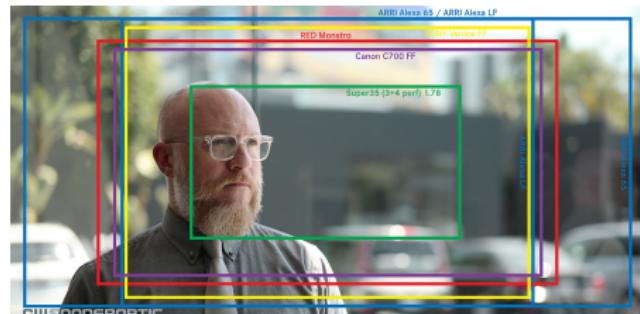
Tomado de courses.cs.washington.edu

Parámetros de cámara *Pinhole*

Tamaño película o del sensor

Define que tanta parte de la escena es capturada

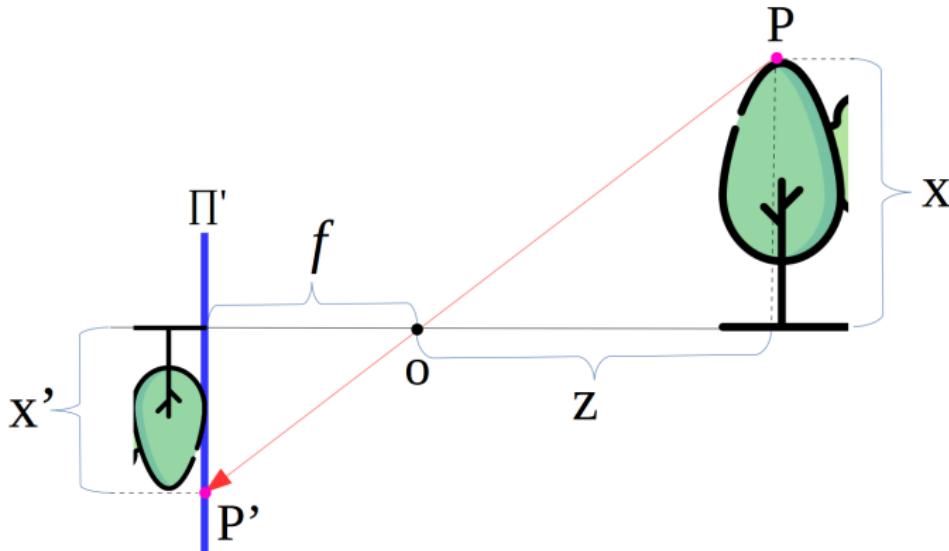
Este parámetro junto a la distancia focal controlan el ángulo de visión



Ángulo de visión con sensores de diferentes tamaños.

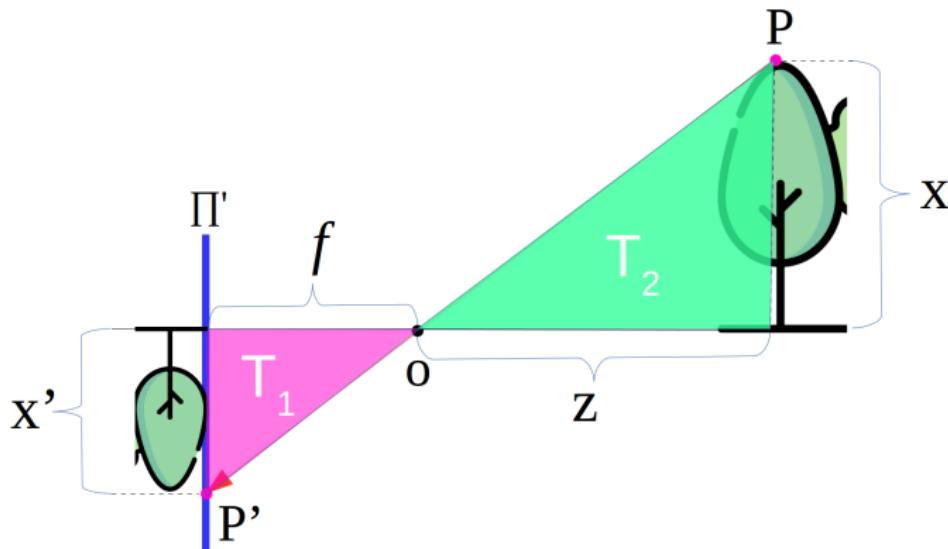
Tomado de ascmag.com

Modelo de cámara *Pinhole* 2D



Determinar la posición del punto P' en el plano de proyección π' de el punto del objeto P , con apertura o y distancia focal f

Modelo de cámara *Pinhole* 2D



El triángulo ΔT_1 es similar al triángulo ΔT_2 , por lo tanto los ratios entre sus ángulos son iguales

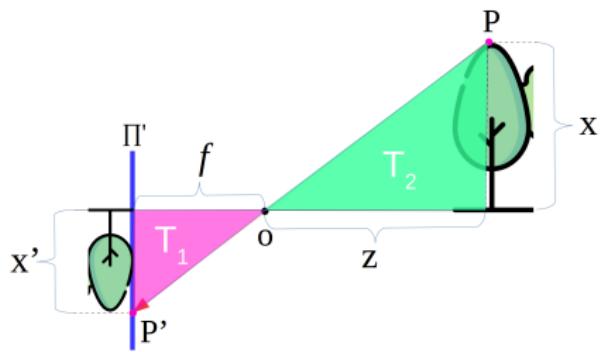
Parámetros de cámara *Pinhole*

Por triángulos similares

$$\frac{x}{z} = \frac{x'}{f}$$

Organizando y definiendo en y

$$x' = f \frac{x}{z}; f \frac{y}{z}$$



Parámetros de cámara *Pinhole*

Como operación matricial en coordenadas homogéneas:

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

Luego al pasar a cartesianas

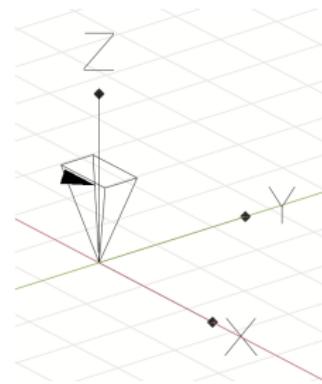
$$[f\frac{x}{z}, f\frac{y}{z}, f]^T$$

Modelo de cámara *Pinhole*

Posición de la cámara

Según las operaciones anteriores
la cámara se encuentra:

- ▶ Centrada en el origen
- ▶ Alineada al eje z



Posición por defecto de la cámara virtual

Modelo de cámara *Pinhole*

Conclusiones

- ▶ Simula el comportamiento ideal de una cámara *Pinhole*
- ▶ No tiene en cuenta distorsiones de lentes o de la apertura
- ▶ Efectos son pequeños y pueden ser compensados
- ▶ Normalmente usada en CG



Architectural Rendering Exterior por Desert Rose

Lentes

Adición de los lentes

- ▶ Perforación reemplazada por los lentes
- ▶ Mitigan el problema entre definición vs. brillo

*Tamaño y posición correcta
satisfacen la propiedad de que todo
punto emitido de la escena converge
a un único punto*



Por Daven Mathies en Digital Trends

Características de los lentes



Tomado de picturecorrect.com

Características Cámaras de lente

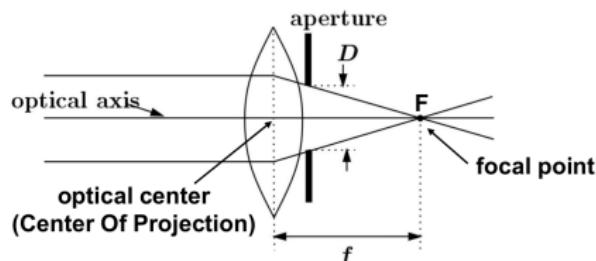
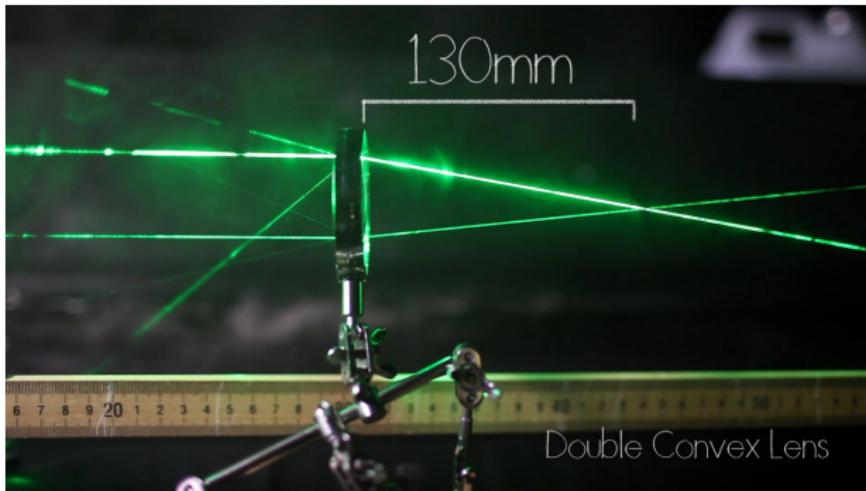


Image Formation and Cameras por Saad J Bedros

- ▶ Rayos que pasan por el centro no son desviados
- ▶ Más lejos del centro más desviación

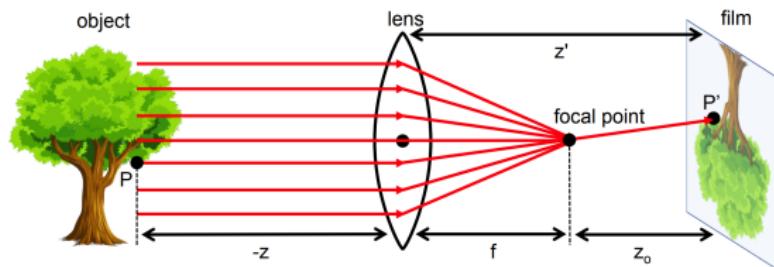
Características Cámaras de lente



The history and science of lens por John P. Hess

Rayos paralelos convergen en un mismo punto

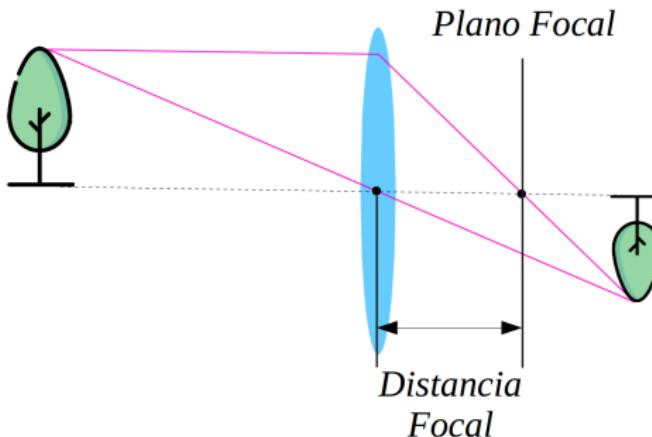
Características Cámaras de lente



Stanford Course CS231A Notes 1: Camera Models por Kenji Hata y Silvio Savarese

El punto en el que convergen los rayos paralelos **eje óptico** es
llamado **Punto Focal**

Características Cámaras de lente

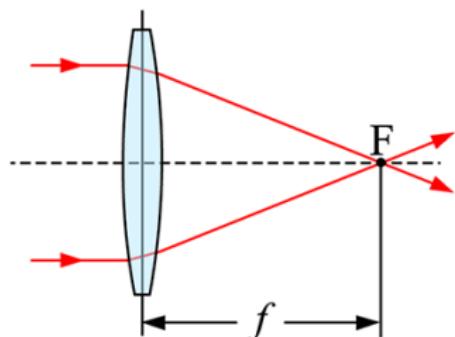


- ▶ Plano paralelo al lente en el punto focal es llamado **Plano focal**
- ▶ Distancia entre el plano y los lentes **Distancia focal**

Características Cámaras de lente

Distancia focal

- ▶ Propiedad depende del índice de refracción y de la forma
- ▶ Que tan fuerte converge la luz
- ▶ Especificado en *mm*

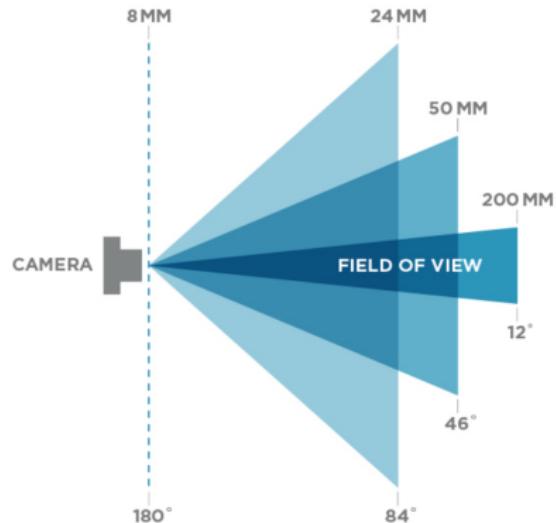


Tomado de howtogeek.com

Características Cámaras de lente

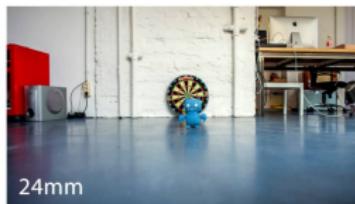
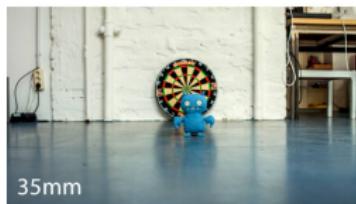
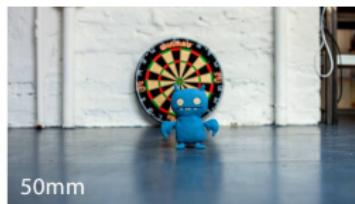
Distancia focal

- ▶ Inversamente proporcional al campo de visión (FOV)
- ▶ Factor de escalamiento



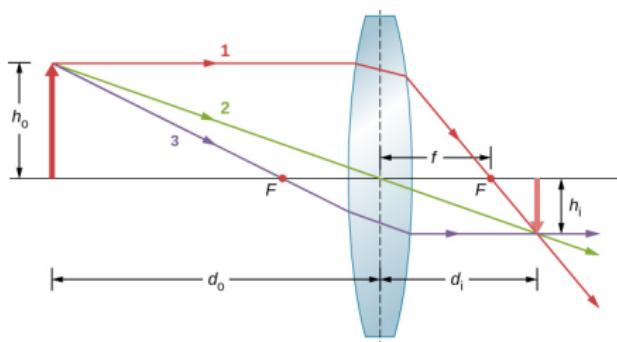
Tomado de blog.cammy.com

Efecto distancia focal



Tomado de s.studiobinder.com

Características Cámaras de lente



Tomado de phys.libretexts.org

Modelado con la suposición de **Thin lens** con aproximación paraxial

Ecuación de *Thin lens*

Ecuación de *Thin lens*

- Relación entre el punto observado \mathbf{P} , el lente y el plano de convergencia Π'
- Puntos que satisfacen esta relación están enfocados

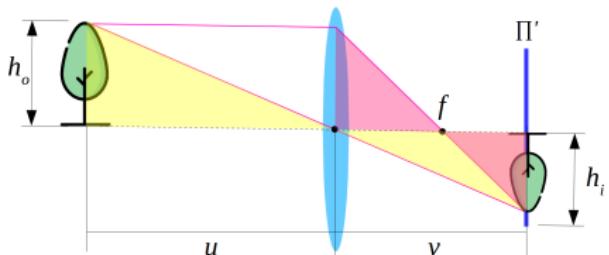


Diagrama de refracción paraxial.

Ecuación de *Thin lens*

Thin lens equation

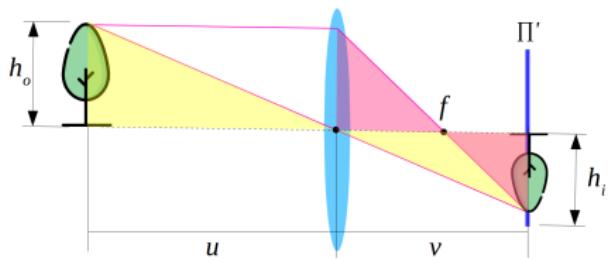
Derivado usando triángulos similares:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

Asumimos que $u \rightarrow \infty$, entonces:

$$v \approx f$$

*Si todos los rayos fueran paralelos y perpendiculares al lente



Ecuación de *Thin lens*

Ecuación de proyección

La relación de punto en el espacio con su proyección

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} z' \frac{x}{Z'} \\ z' \frac{y}{Z'} \end{bmatrix}$$

Similar a la ecuación *pinhole* ($z' = f$),
solo que
 $z' \approx f$

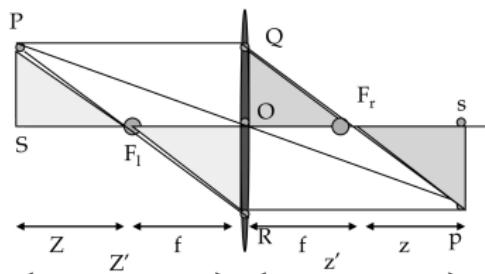


Diagrama de refracción paraxial. Tomado de *Cameras and Image Formation* por Ahmed Elgammal

Camera matrix

Centro coordenado a esquina izquierda superior

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f \frac{x}{z} + c_x \\ f \frac{y}{z} + c_y \end{bmatrix}$$

Cambio de unidades mm a px (factores k y l)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} fk \frac{x}{z} + c_x \\ fl \frac{y}{z} + c_y \end{bmatrix} = \begin{bmatrix} \alpha \frac{x}{z} + c_x \\ \beta \frac{y}{z} + c_y \end{bmatrix}$$

Expresado en coordenadas homogéneas

$$\begin{bmatrix} x' \\ y' \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z \\ 1 \end{bmatrix}$$

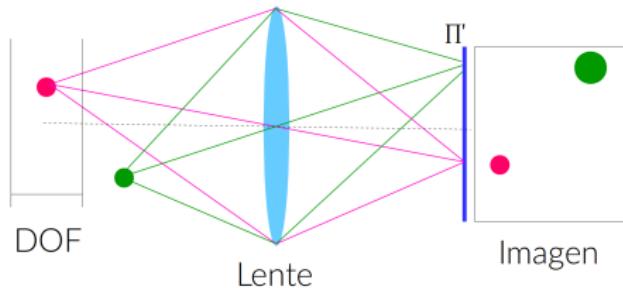


Si todos convergen a un punto ¿qué pasó con estos?
Tomado de nikondigital.org

Cámaras de lente

Profundidad de campo

- ▶ Depth of Field (DOF)
- ▶ Puntos de la escena convergen a distancias diferentes
- ▶ Objetos dentro del *DOF* están enfocados
- ▶ Controlado con: **Apertura** y **Distancia focal**
- ▶ dofsimulator.net/en

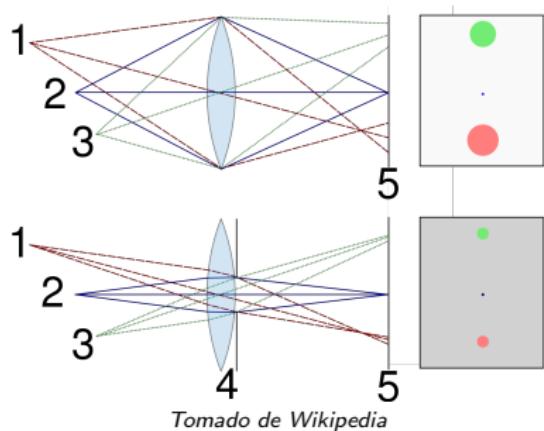


Profundidad de campo dependiendo de la distancia de los objetos

Profundidad de campo

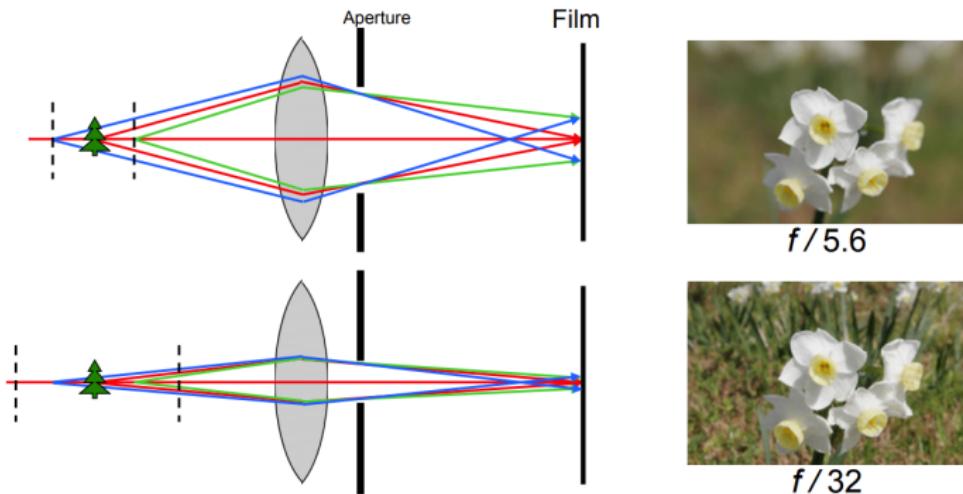
Apertura

- ▶ Cambiando la apertura cambia la profundidad de campo
- ▶ Aperturas más pequeñas:
 - ▶ Mayor rango enfocado
 - ▶ Menor intensidad



Apertura es el agujero por el cual ingresa la luz.
Determina el ángulo del cono dirigido al sensor

Profundidad de campo



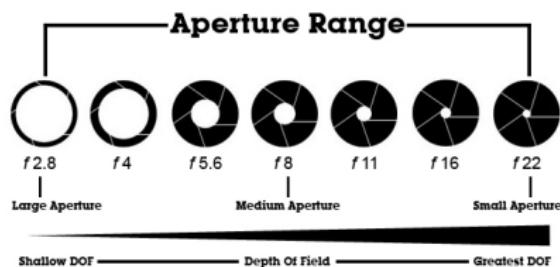
Cambiando la apertura cambia el rango en que los objetos están enfocados.

Tomado de Image Formation and Cameras por Saad J Bedros

Profundidad de campo

F-Stop

- ▶ En las cámaras se especifica con *f-stop*
- ▶ Ratio entre distancia focal y apertura
- ▶ Más alto el valor → menor apertura



Tomado de photography.tutsplus.com

Formación de imágenes

Cámaras y lentes

Modelo Cámara Pinhole

Lentes

Thin lenses

Profundidad de campo

De 3D a imagen digital

Transformaciones de sistemas coordinados

Flujo de transformaciones

Determinación de caras visibles

Back-face elimination

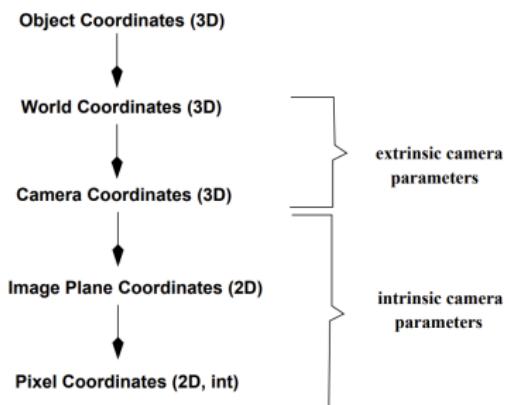
Depth-Sort

Z-Buffer

Bibliografía

Flujo de sistemas de referencia

Objetos	Dónde se modelan los objetos
Mundo	Modelos puestos en la escena
Cámara	Objetos respecto a la ubicación de la cámara
Plano de imagen	Proyección 2D
Pixeles	Imagen rasterizada



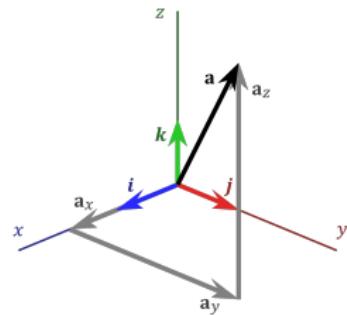
Reference frames. Tomado de cse.unr.edu

Transformaciones de Sistemas Coordenados

Sistemas Coordenados

- ▶ Los **vectores** $[x, y, z]^T$ representan distancias desde el origen en un eje
- ▶ **Sistema Coordenado** es la referencia con la cual son medidos
- ▶ Puede ser representado con los vectores unitarios (**vectores bases**)

$$\hat{\mathbf{i}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{\mathbf{j}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \hat{\mathbf{k}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Standard basis. Tomado de Wikipedia

Transformaciones de Sistemas Coordenados

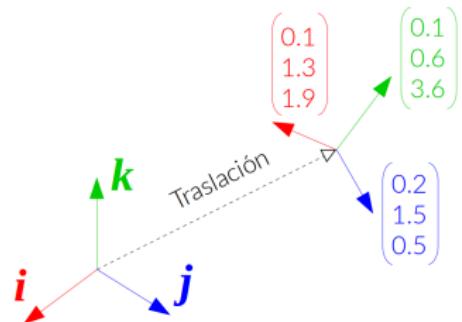
Transformación Sistemas Coordenados

- ▶ Transformar sus **vectores bases**
- ▶ Representado con una matriz

Donde terminan

Traslación

$$\begin{matrix} i & j & k \\ \begin{pmatrix} 0.1 & 0.2 & 0.1 & 1.3 \\ 1.3 & 1.5 & 0.6 & 2.5 \\ 1.9 & 0.5 & 3.6 & 3.6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$



Transformar objetos de sistemas coordinados

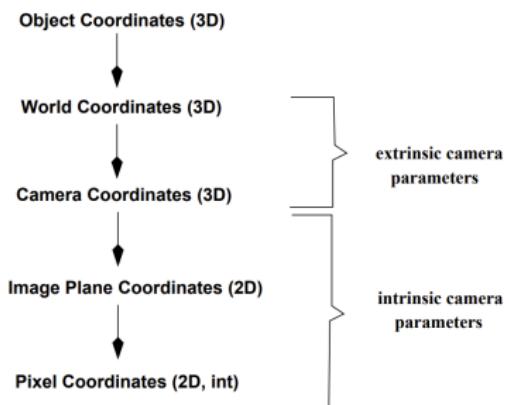
No transformamos el sistema coordenado. Aplicamos la transformación a los vértices del objeto

$$P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4 \times 4} P_w$$

$$\begin{pmatrix} v'_{1x} & v'_{2x} & \dots & v'_{nx} \\ v'_{1y} & v'_{2y} & \dots & v'_{ny} \\ v'_{1z} & v'_{2z} & \dots & v'_{nz} \\ 1 & 1 & \dots & 1 \end{pmatrix} = \begin{pmatrix} 0.1 & 0.2 & 0.1 & 1.3 \\ 1.3 & 1.5 & 0.6 & 2.5 \\ 1.9 & 0.5 & 3.6 & 3.6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_{1x} & v_{2x} & \dots & v_{nx} \\ v_{1y} & v_{2y} & \dots & v_{ny} \\ v_{1z} & v_{2z} & \dots & v_{nz} \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

Flujo de sistemas de referencia

Objetos	Dónde se modelan los objetos
Mundo	Modelos puestos en la escena
Cámara	Objetos respecto a la ubicación de la cámara
Plano de imagen	Proyección 2D
Pixeles	Imagen rasterizada

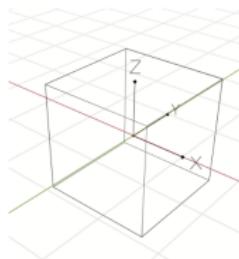


Reference frames. Tomado de cse.unr.edu

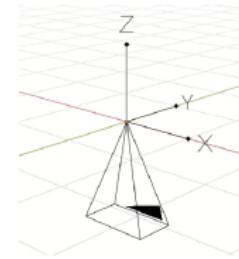
Flujo de sistemas de referencia

Sistema de los objetos

- ▶ Sistema coordenado dónde se modelan los objetos 3D.
- ▶ Cámara es ubicada en el origen en dirección z



Objetos

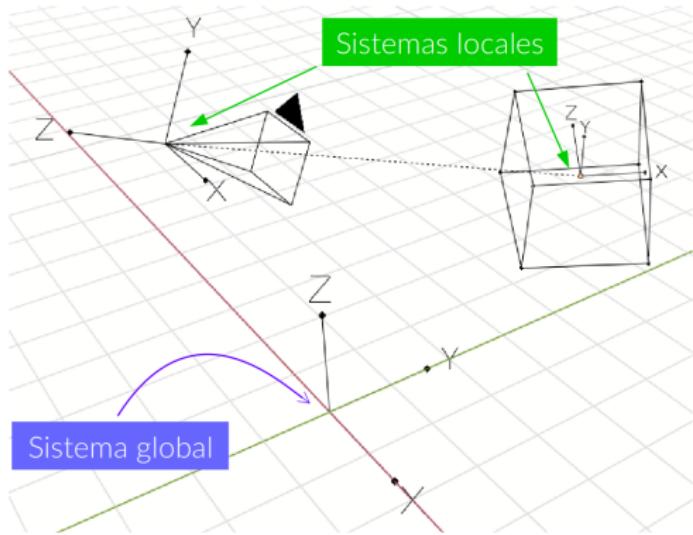


Cámara

Flujo de sistemas de referencia

Sistema del mundo o global

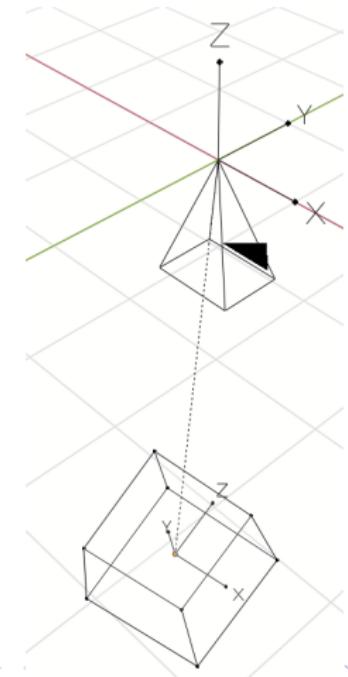
- ▶ Sistema coordenado de la escena
- ▶ Son posicionados los objetos y la cámara
- ▶ Sistema coordenado cartesiano



Flujo de sistemas de referencia

Sistema local de la cámara

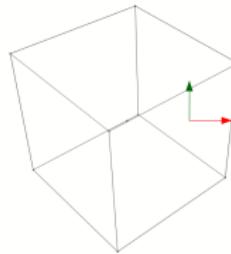
- ▶ Transformar objetos al sistema local de la cámara
 - ▶ Matriz inversa de la transformación de la cámara a global
- $$P' = M_G^{-1}P$$
- ▶ **Parámetros extrínsecos de la cámara**



Flujo de sistemas de referencia

Sistema coordinado del plano de proyección

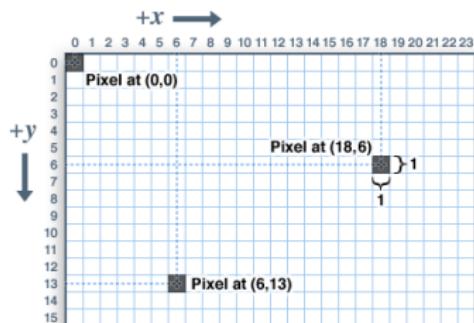
- ▶ Proyectamos usando el modelo *Pinhole* ó de *Thin lens*
- ▶ Espacio continuo
- ▶ Origen en el centro del la imagen
- ▶ **Parámetros intrínsecos de la cámara**



Flujo de sistemas de referencia

Espacio Rasterizado

- ▶ Espacio discreto (pixeles)
- ▶ Centro coordenado en la esquina izquierda
- ▶ Proyección intermedia (NCD)

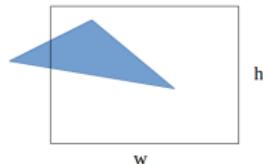


Espacio rasterizado. Tomado de learn.adafruit.com

Espacio Rasterizado

1. Cortar a tamaño de la image

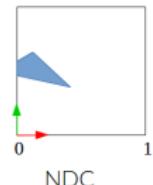
$$|P'_x| \leq \frac{w}{2} \text{ or } |P'_y| \leq \frac{h}{2} \rightarrow \text{visible}$$



Plano de proyección

2. Normalizar $[0,1]$ coordenadas

$$(P'_x, P'_y) = \left[\frac{P'_x + w/2}{w} \quad \frac{P'_y + h/2}{h} \right]$$



NDC

3. Multiplicar por tamaño de pixel (pw, ph)

$$(P'_x, P'_y) = (\lfloor P'_x pw \rfloor, \lfloor P'_y ph \rfloor)$$

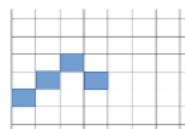
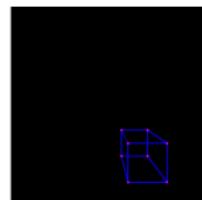


Imagen rasterizada

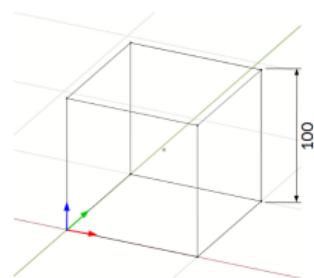
Ejercicio Proyecciones

Definir:

- ▶ Geometría planteada:
 - ▶ Vértices
 - ▶ Bordes
- ▶ Función de traslación 3D
- ▶ Función de proyección tipo *Pinhole*
- ▶ Trasladar cubo a
 $x = 100, y = 200, z = 50$
- ▶ Trasladar proyección al centro



Resultado Final



Geometría

Formación de imágenes

Cámaras y lentes

Modelo Cámara Pinhole

Lentes

Thin lenses

Profundidad de campo

De 3D a imagen digital

Transformaciones de sistemas coordenados

Flujo de transformaciones

Determinación de caras visibles

Back-face elimination

Depth-Sort

Z-Buffer

Bibliografía

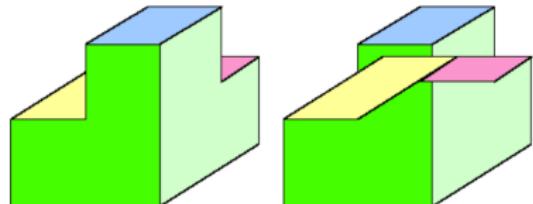
Determinación de caras visibles

Problema

Identificar que superficies son visibles desde un punto de vista

Nombres

- ▶ Determinación de las caras ocultas
- ▶ Hidden-surface determination
- ▶ Occlusion culling (OC)
- ▶ Visible-surface determination (VSD)

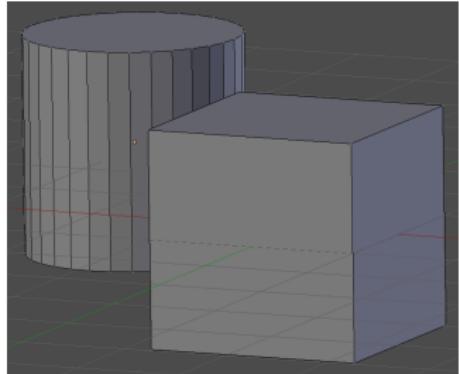
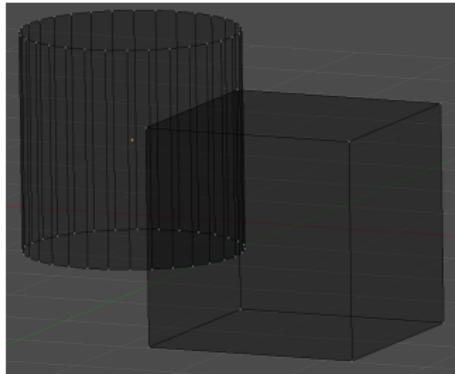


Tomado de glasnost.itcarlow.ie

Determinación de caras visibles

Causas

- ▶ Oclusión por otros elementos
- ▶ Oclusión por si mismo



Determinación de caras visibles

Algoritmos

Ideas simples

- ▶ Back-face elimination
- ▶ Depth-Sort (painter)

Ideas complejas

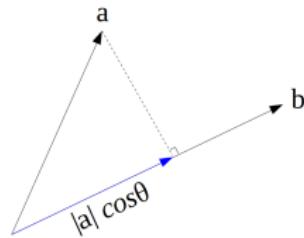
- ▶ BSP-trees
- ▶ Scanline
- ▶ Z-Buffer
- ▶ Raytracing

Determinación de caras visibles

Back-face elimination

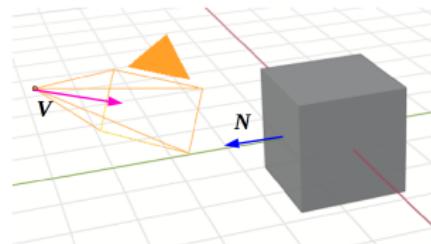
Normales de las caras visibles apuntan a la cámara.

$$N \cdot V < 0 : \text{visible}$$



$$a \cdot b = |a| |b| \cos(\theta)$$

Definición de producto punto



Vectores en escena

Determinación de caras visibles

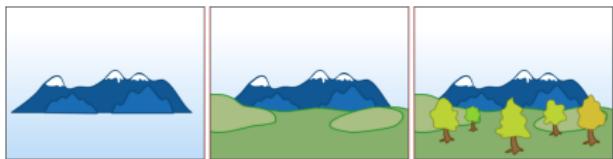
Back-face elimination

- ▶ Rápido y bajos recursos
- ▶ Usado como paso preliminar
- ▶ Suficiente para un objeto convexo

Determinación de caras visibles

Depth-Sort (painter)

- ▶ Dibujar polígonos que están lejos primero
- ▶ Polígonos cercanos ocultan los anteriores
- ▶ Similar a pintar en la vida real

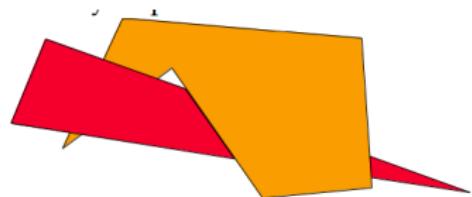


Painter algorithm tomado de Wikipedia

Determinación de caras visibles

Depth-Sort (painter)

1. Ordenar polígonos por su distancia a la cámara (vértice más cercano)
2. Comenzar desde el más lejano
3. Identificar conflictos (rangos interpolados)

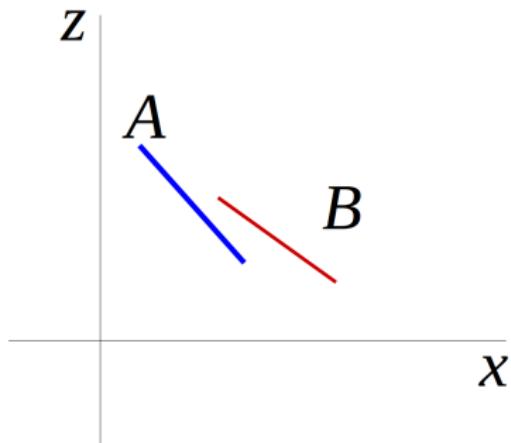


Conflictos Painter algorithm tomado de cis.temple.edu

Depth-Sort (painter)

Conflictos

- ▶ El z más cercano de B es menor que el de A
- ▶ B tiene que ser renderizado primero
- ▶ ¿Por qué?
- ▶ En este caso deben ser reorganizados

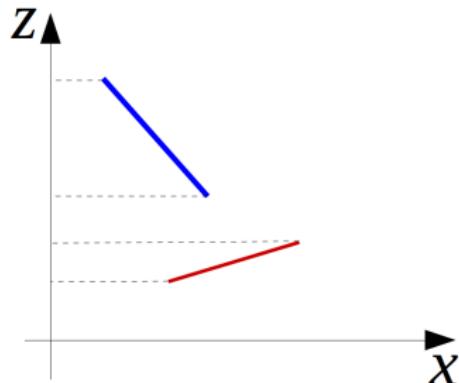


Depth-Sort (painter)

Resolución del conflicto

Pueden ser reordenados si:

1. Proyección en z no se sobrepone

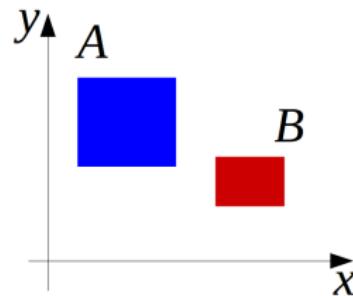


Depth-Sort (painter)

Resolución del conflicto

Pueden ser reordenados si:

2. *Bounding box* en el plano XY no se sobreponen

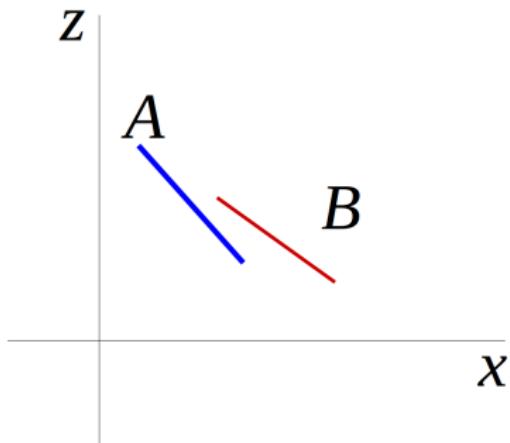


Depth-Sort (painter)

Resolución del conflicto

Pueden ser reordenados si:

3. Está completamente atrás del otro con respecto el punto de vista



Determinación de caras visibles

Z-Buffer

- ▶ También llamada *Depth buffering*
- ▶ Más simple y usada, (todos los computadores)
- ▶ Tarea realizada en la tarjeta de video
- ▶ Descrita en 1974 por Wolfgang Straßer en su PhD

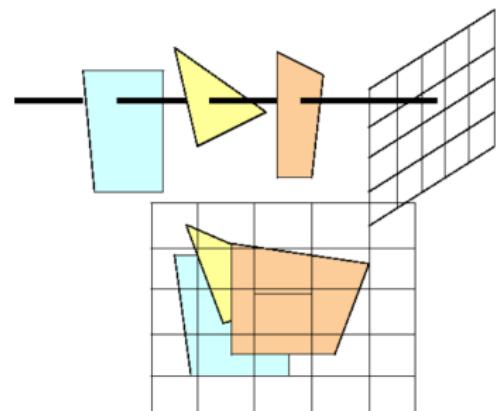


Mapa de profundidad. Tomado de teliaoils.com

Z-Buffer

Idea

- ▶ Definir la visibilidad pixel a pixel
- ▶ El color de cada pixel corresponde al punto de la superficie con menor z

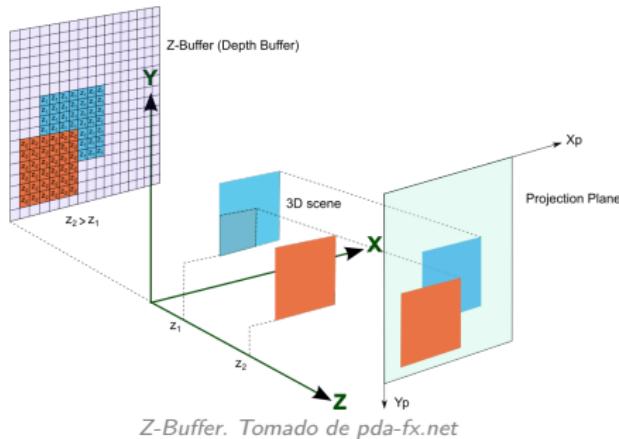


Z-Buffer. Tomado de glasnost.itcarlow.ie

Z-Buffer

Idea

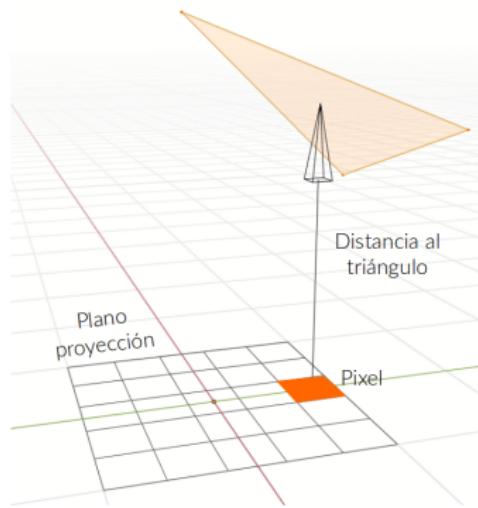
- ▶ Iniciar con 2 *buffers*: Profundidad y Color
- ▶ Por cada pixel de cada polígono proyectado:
 - ▶ Calcular profundidad z
 - ▶ Si es menor actualizar *buffers*



Z-Buffer

Retos

1. Identificar si punto (x, y) está en el triángulo
2. Calcular la distancia a la cámara



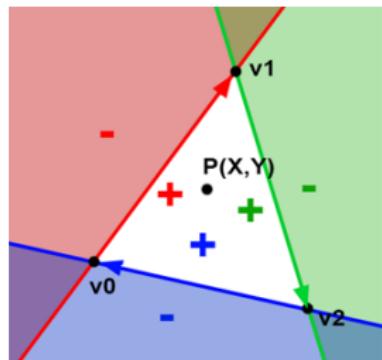
Z-Buffer

Inside-Outside Test

- ▶ Un punto está dentro/fuera de un triángulo
- ▶ **Edge Function** por Juan Pineda (1988)

Idea

Función que **determine el lado** en el que se encuentre un punto (P) con **respecto a un vector** (V)



Edge function. Tomado de scratchapixel.com

Verificar borde por borde del triángulo

Inside-Outside Test

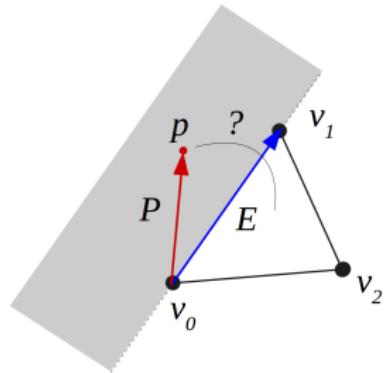
Edge Function

- ▶ **Determinante** de los vectores formados por $(\vec{P} \text{ y } \vec{E})$
- ▶ Valores positivos → Lado derecho

Determinante en vectores

Área del paralelogramo formado sus vectores:

$$\begin{vmatrix} \vec{P}_x & \vec{P}_y \\ \vec{E}_x & \vec{E}_y \end{vmatrix} = \vec{P}_x \vec{E}_y - \vec{P}_y \vec{E}_x$$



Edge Function

Determinante de los vectores es:

$$F_{01}(P) = \vec{P}_x \vec{E}_y - \vec{P}_y \vec{E}_x$$

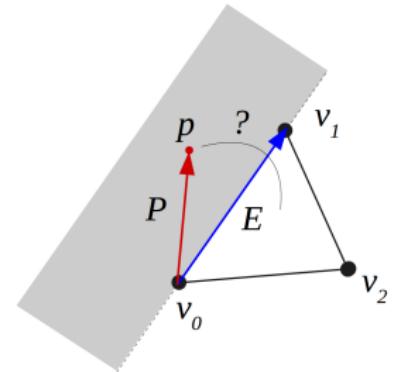
Definiendo los vectores dado sus puntos:

$$F_{01}(P) =$$

$$(p_x - v0_x)(v1_y - v0_y) - (p_y - v0_y)(v1_x - v0_x)$$

De modo que:

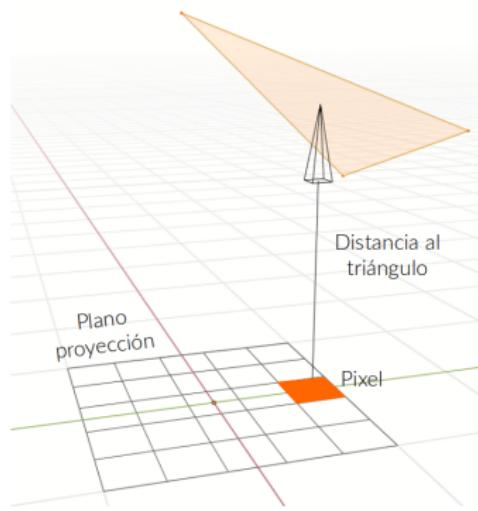
- ▶ $F(P) > 0 \rightarrow$ Lado derecho (adentro)
- ▶ $F(P) < 0 \rightarrow$ Lado izquierdo
- ▶ $F(P) = 0 \rightarrow$ Borde



Z-Buffer

Retos

1. Identificar si punto (x, y) está en el triángulo
2. **Calcular la distancia a la cámara**



Z-Buffer

Distancia a la cámara

Plano está definido por:

$$Ax + By + Cz + D = 0 \text{ donde: } N = \langle A, B, C \rangle$$

Triángulo se encuentra en el plano, la distancia hacia la cámara z dado x, y es:

$$z = \frac{-Ax - By - D}{C}$$

Ejercicio caras visibles

Ejercicio

1. Proyectar el triángulo definido por los vértices
2. Definir la función que determine lado del punto
3. Definir una función que determine si el punto está dentro del triángulo
4. Colorear el interior del triángulo

Bibliografía

Referencias

- ▶ Computer Graphics Using OpenGL por F S. Hill Jr
- ▶ Stanford Course CS231A Notes 1: Camera Models por Kenji Hata y Silvio Savarese