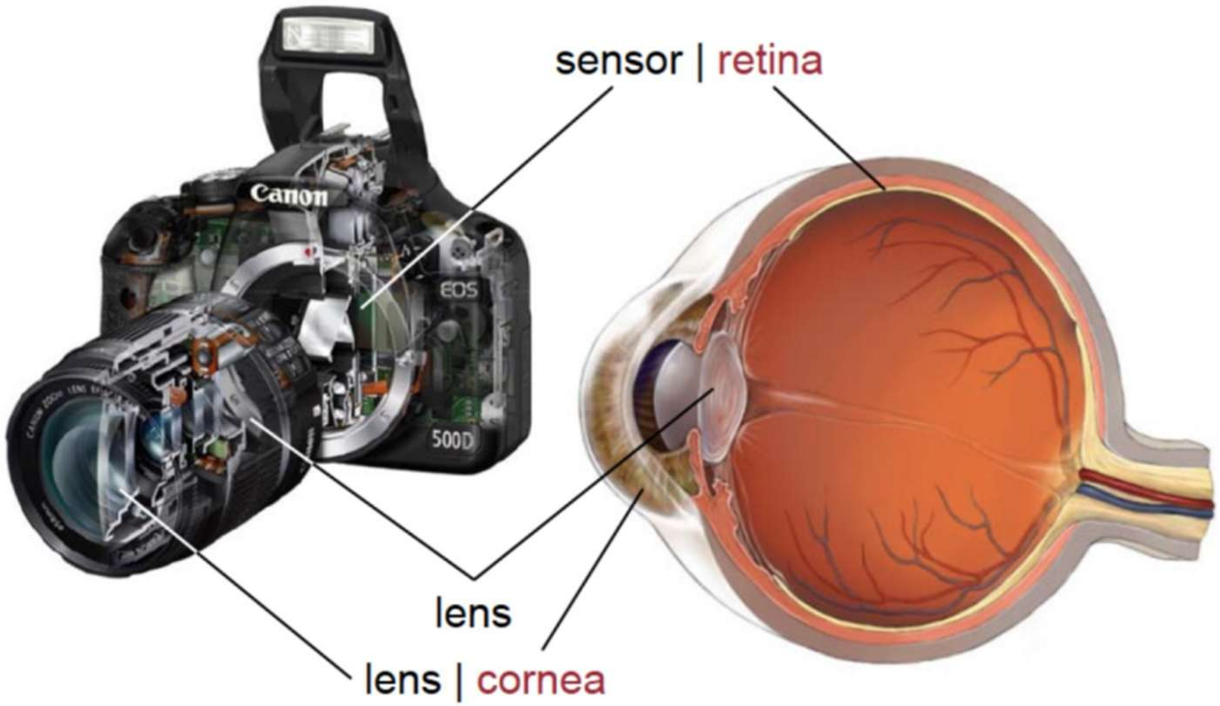
Visión Computacional

Ivan Sipiran

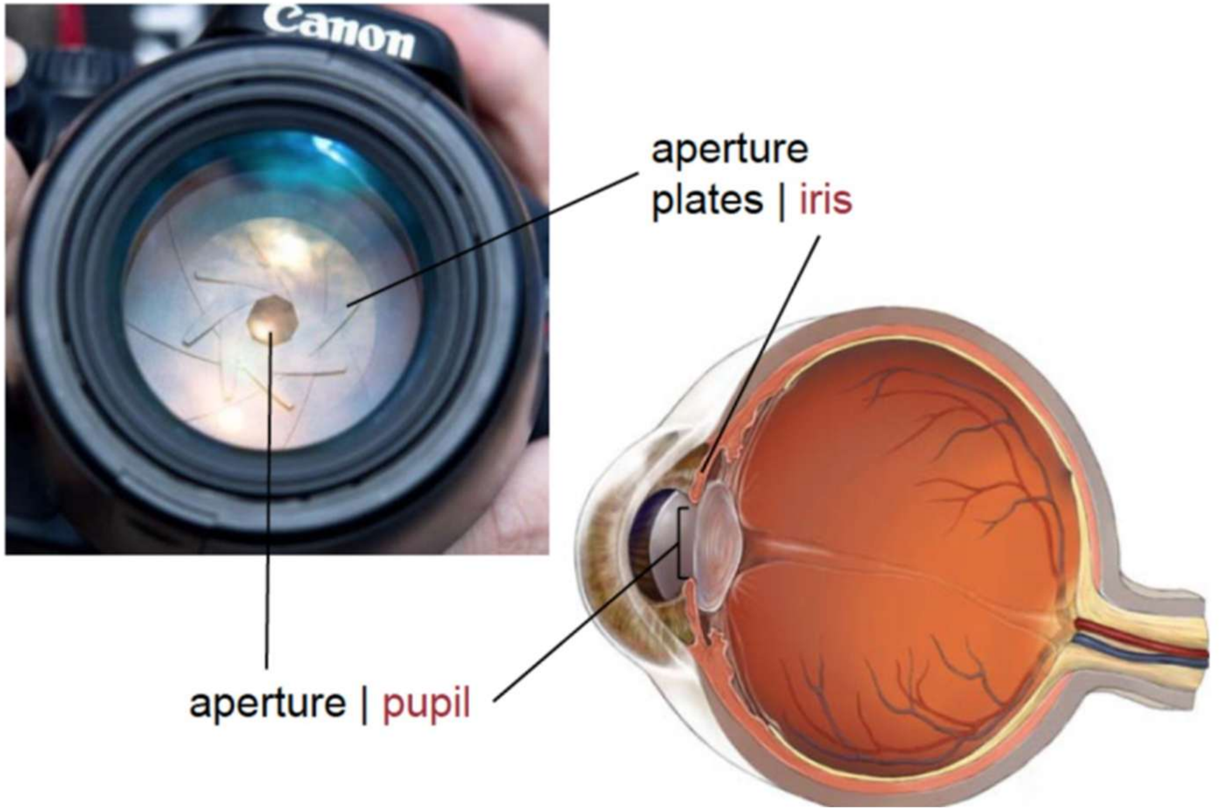
Cámara

Estructuralmente similar a un ojo



Cámara

Estructuralmente similar a un ojo



Cámara Pinhole

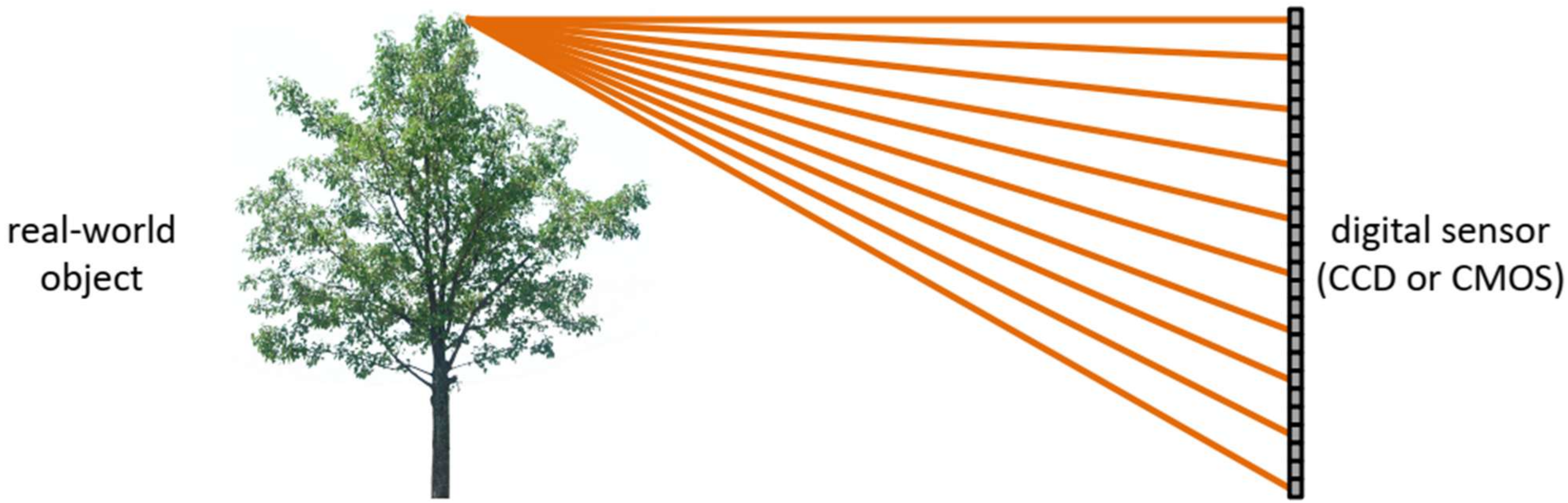
Cómo sería una imagen tomada

Cámara Pinhole

Cómo sería una imagen tomada

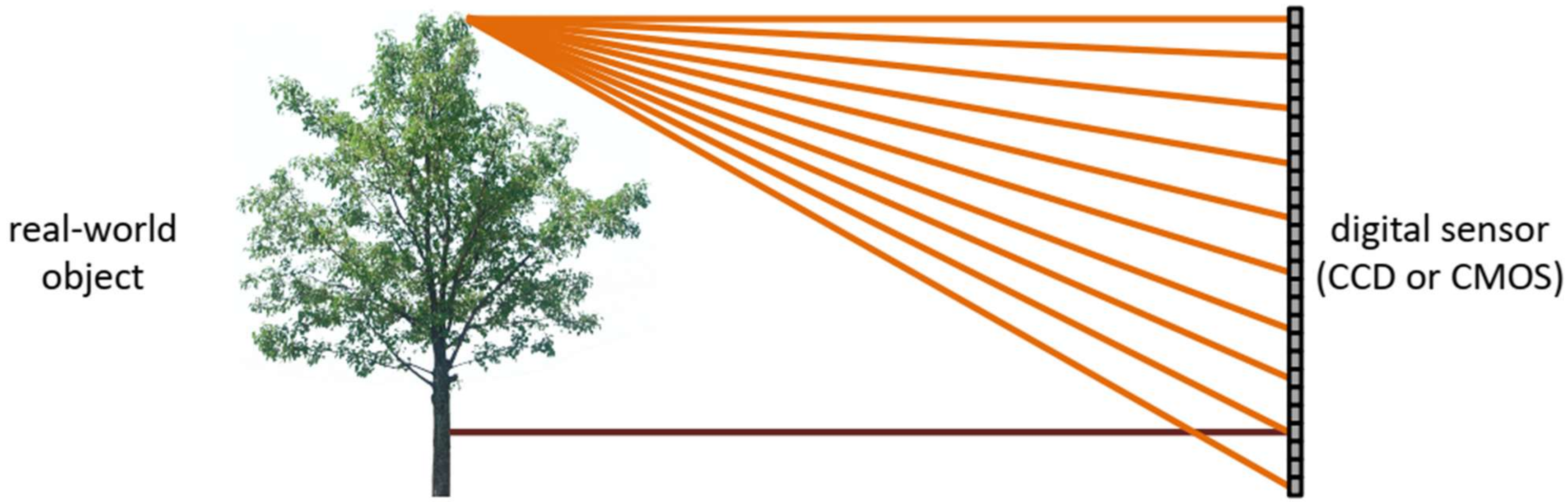
Cámara Pinhole

Cómo sería una imagen tomada



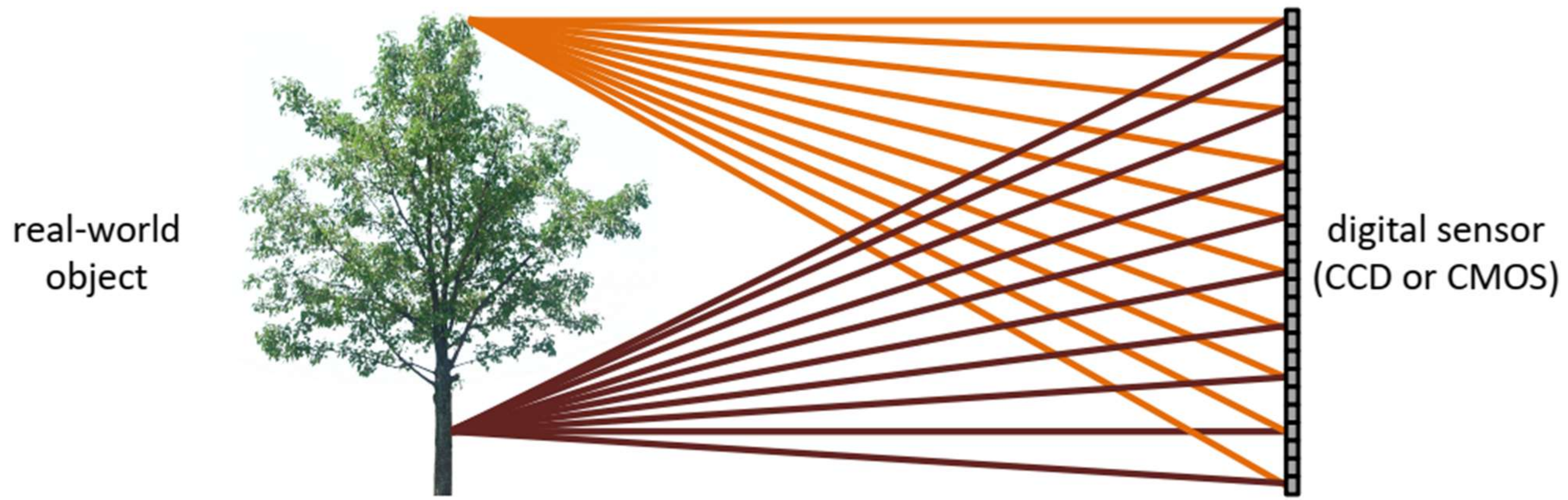
Cámara Pinhole

Cómo sería una imagen tomada



Cámara Pinhole

Todos los puntos de la escena contribuyen a todos los píxels del sensor

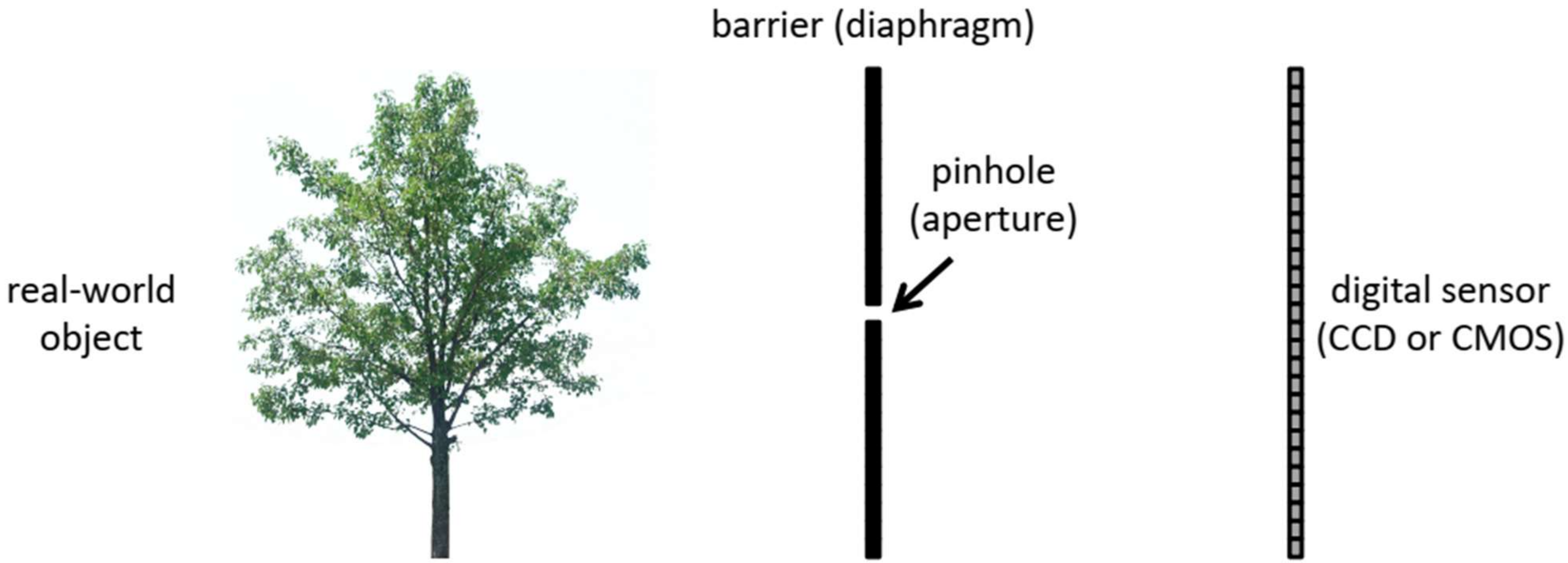


Cámara Pinhole

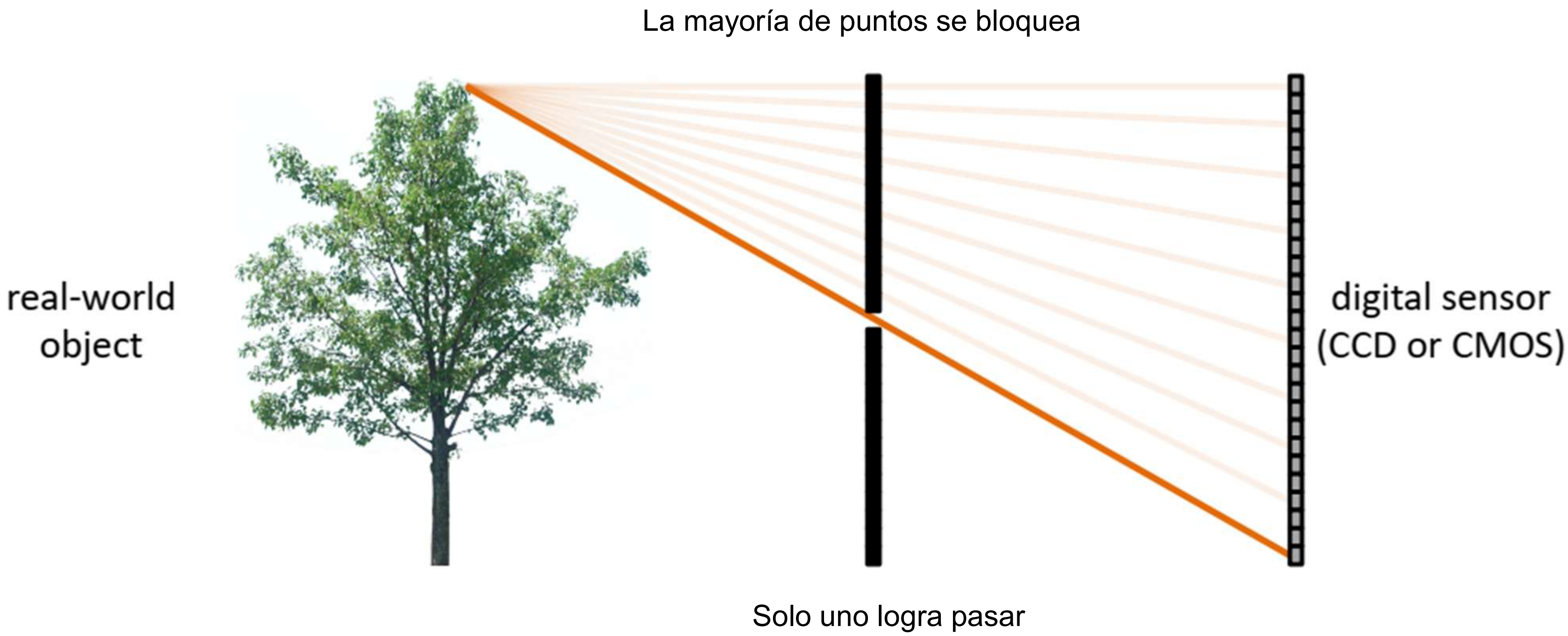
Todos los puntos de la escena contribuyen a todos los píxels del sensor

Cámara Pinhole

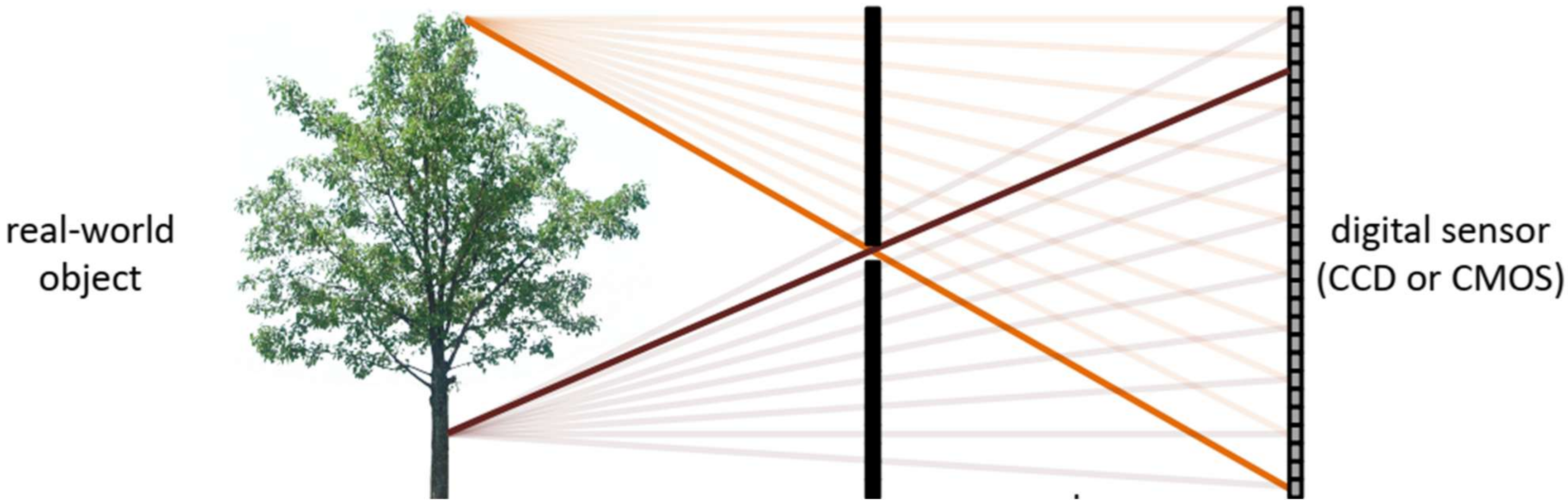
Cámara Pinhole



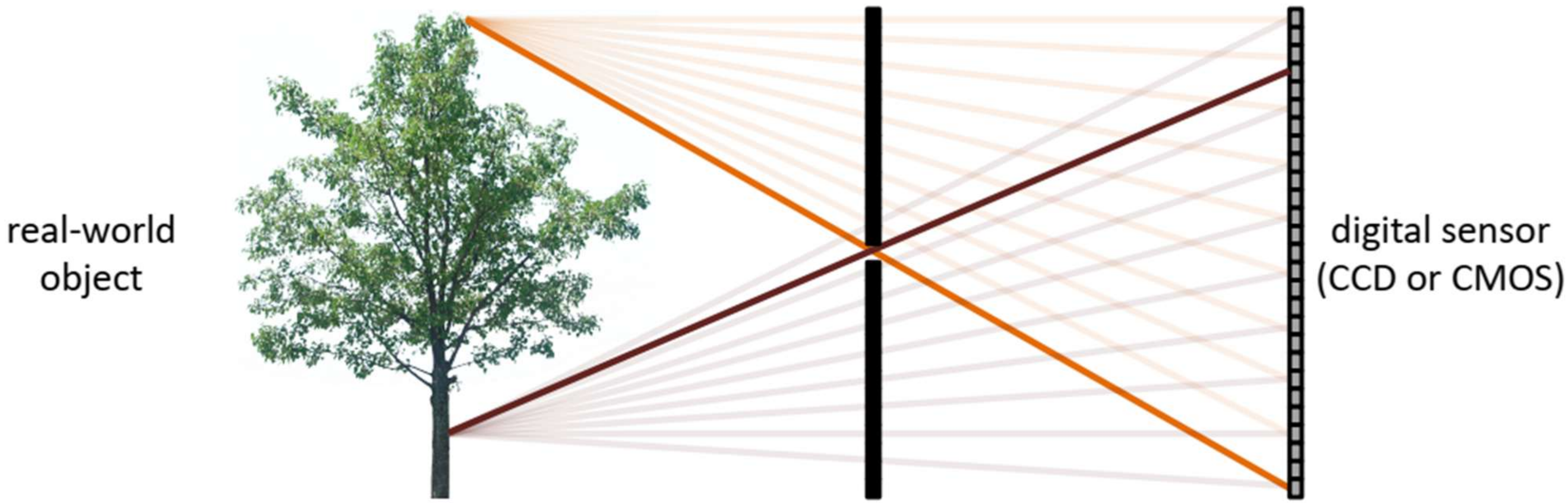
Cámara Pinhole



Cámara Pinhole

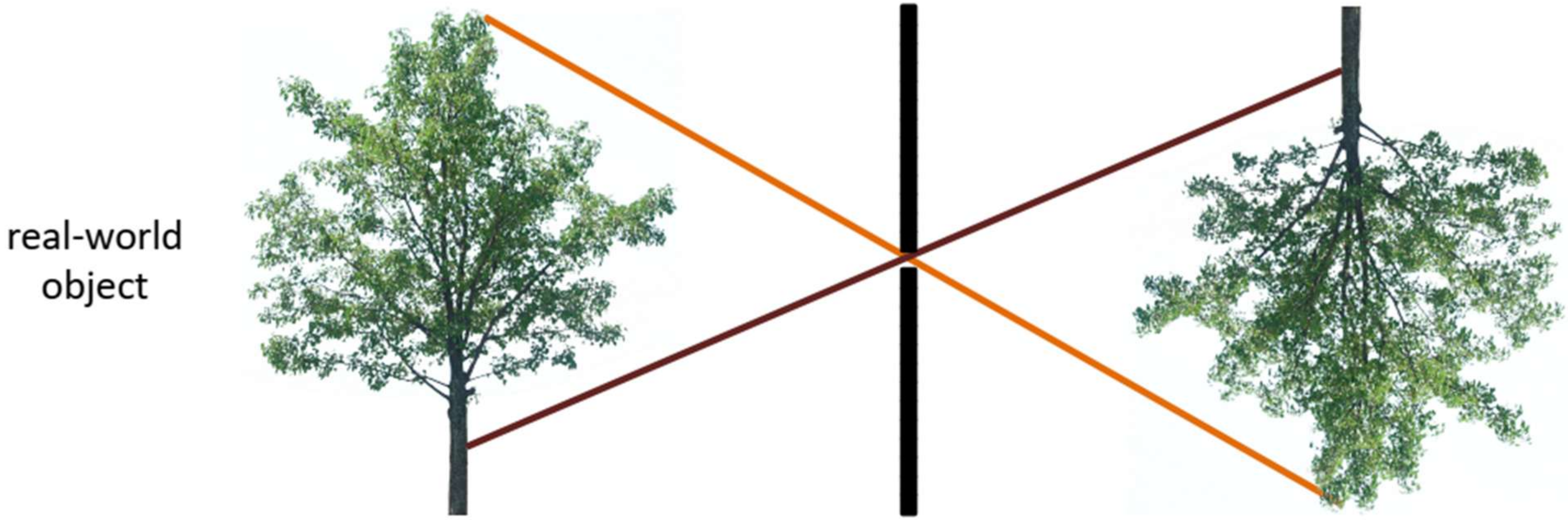


Cámara Pinhole



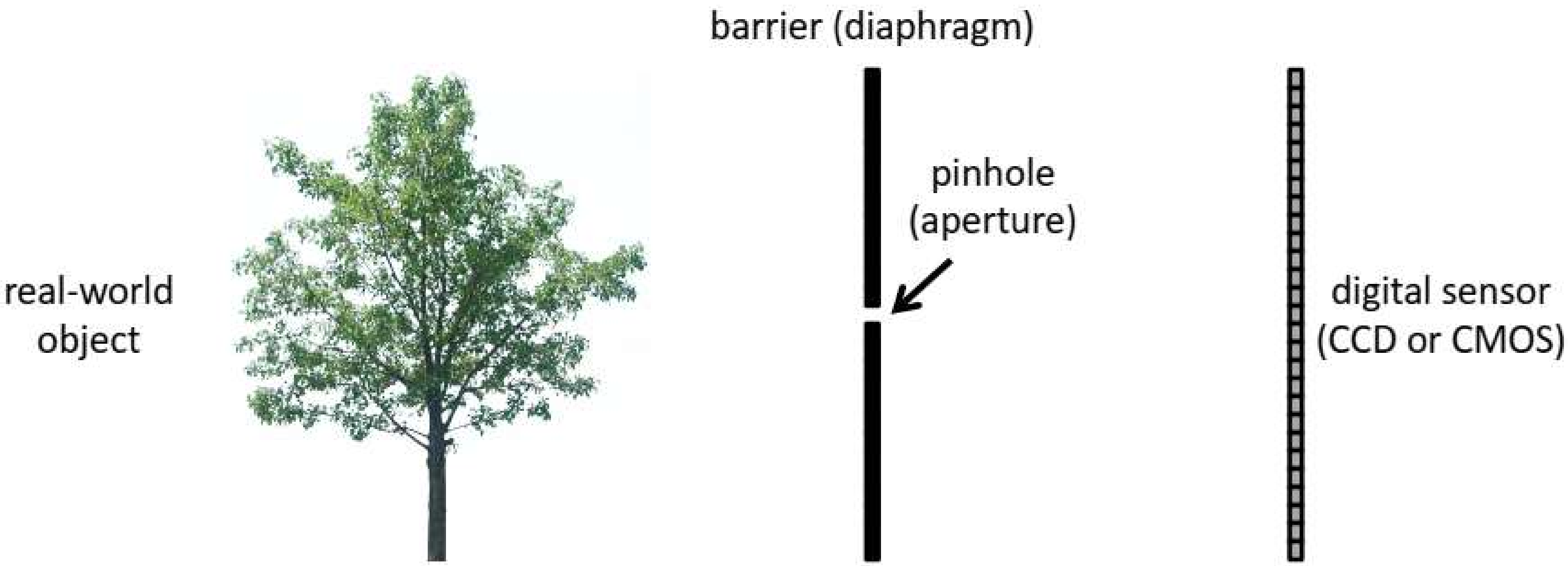
Cada punto de la escena contribuye a un solo punto en la imagen

Cámara Pinhole

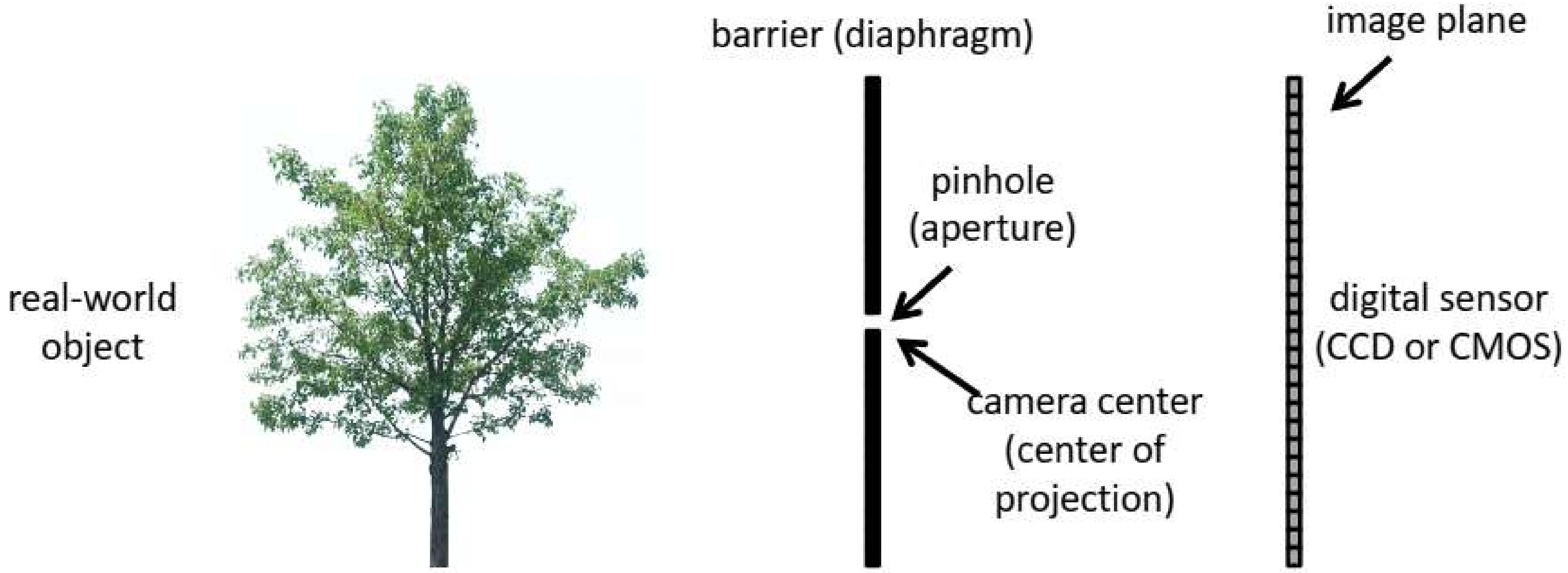


Cada punto de la escena contribuye a un solo punto en la imagen

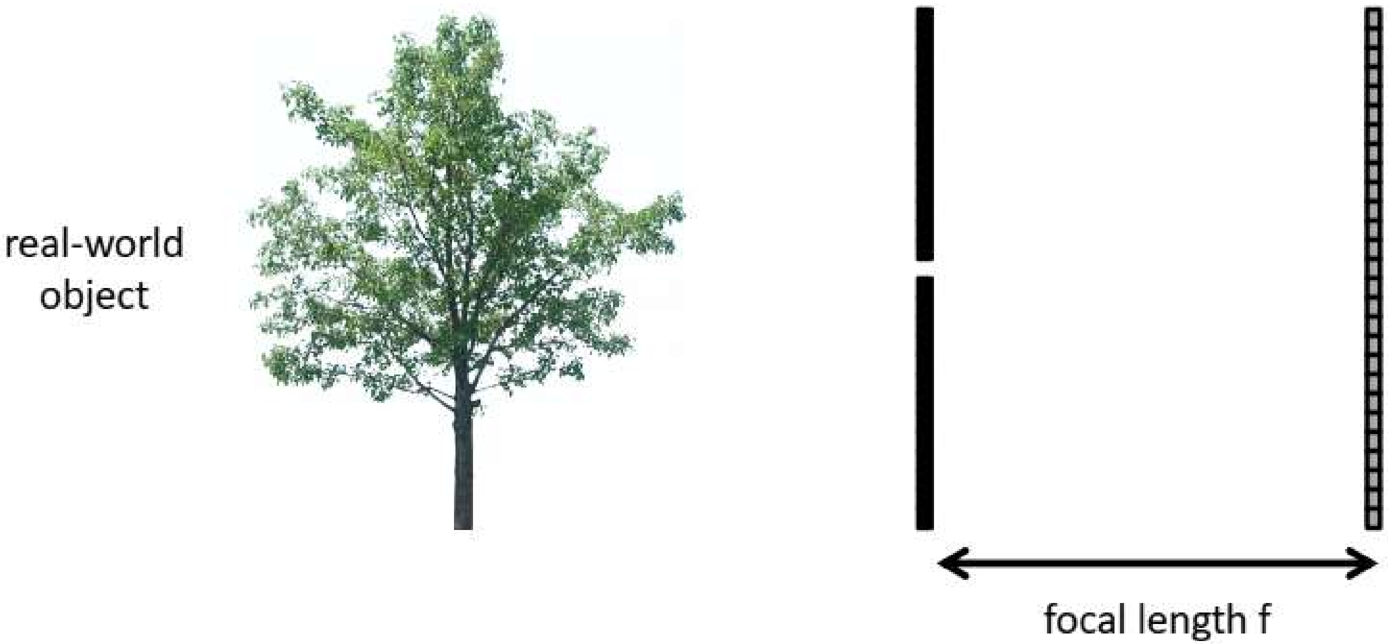
Cámara Pinhole



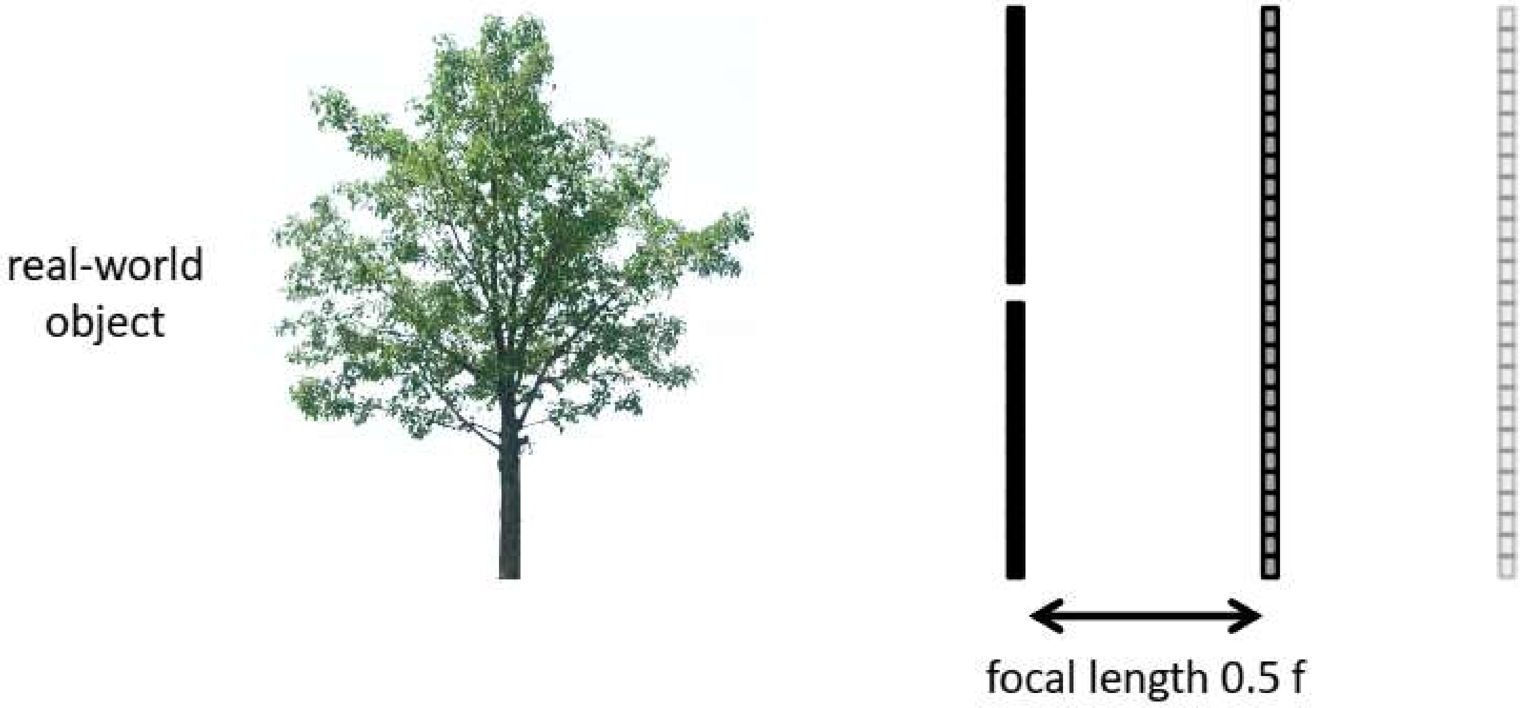
Cámara Pinhole



Cámara Pinhole

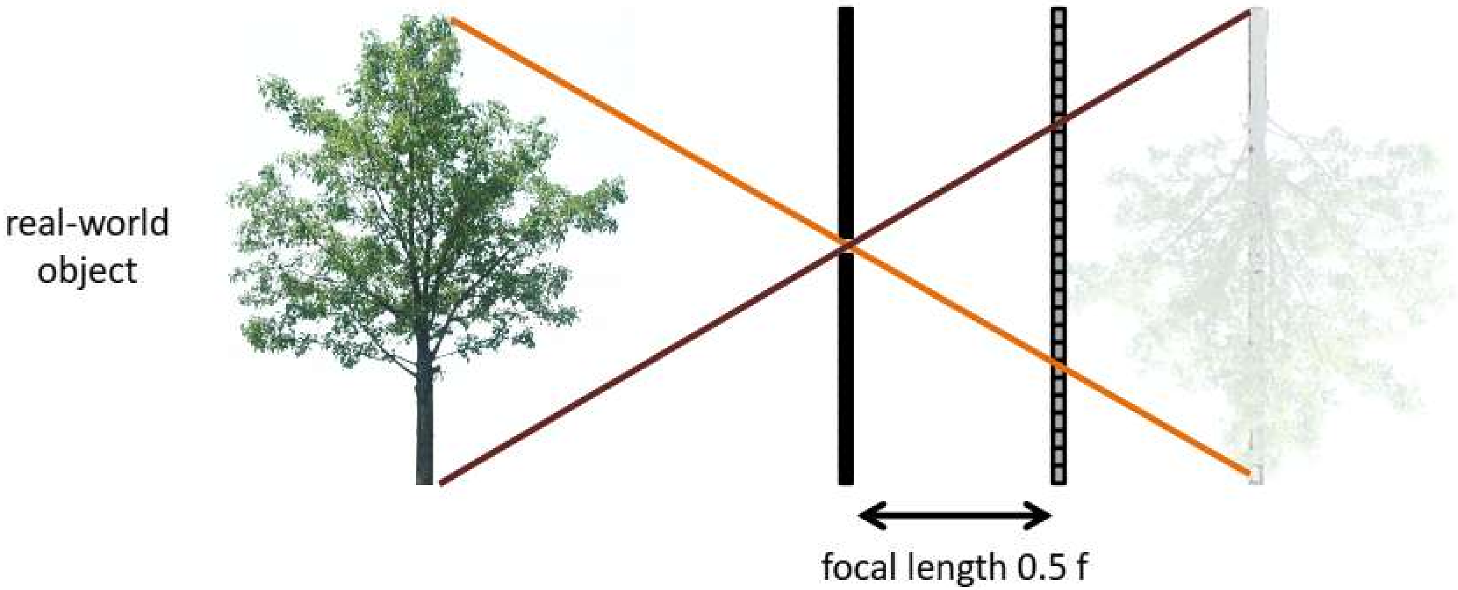


Cámara Pinhole



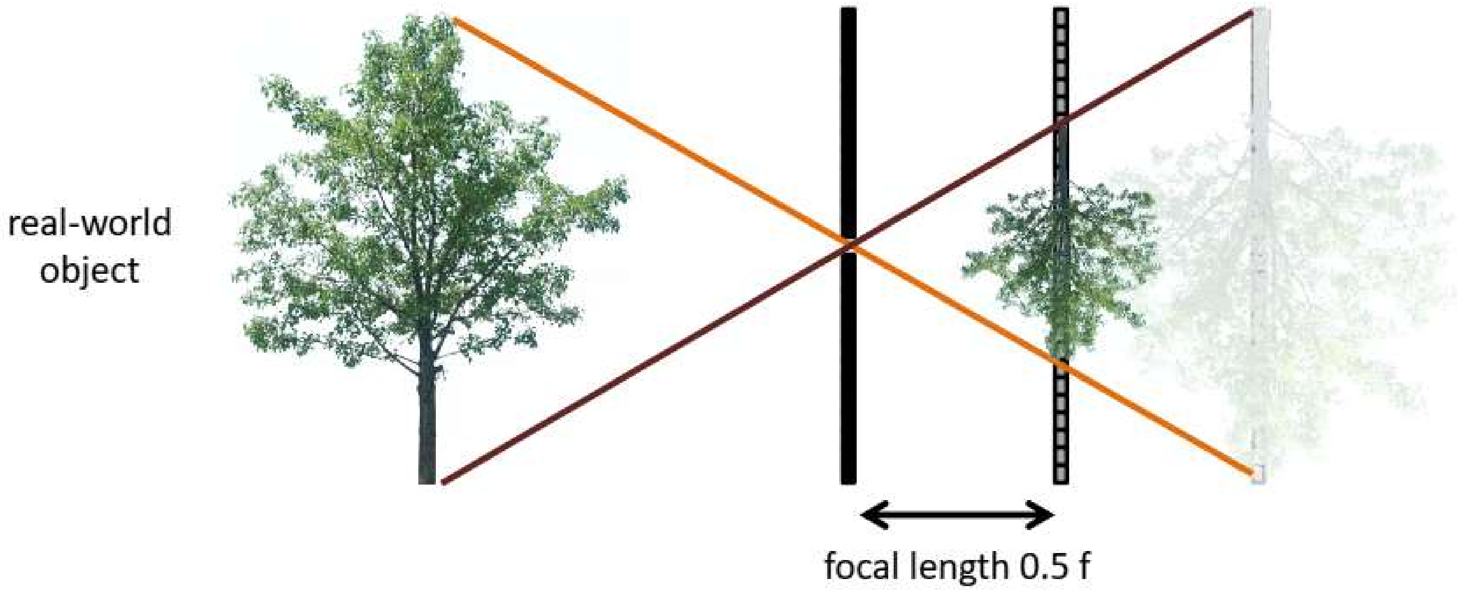
Qué pasa si cambiamos la distancia focal?

Cámara Pinhole



Qué pasa si cambiamos la distancia focal?

Cámara Pinhole

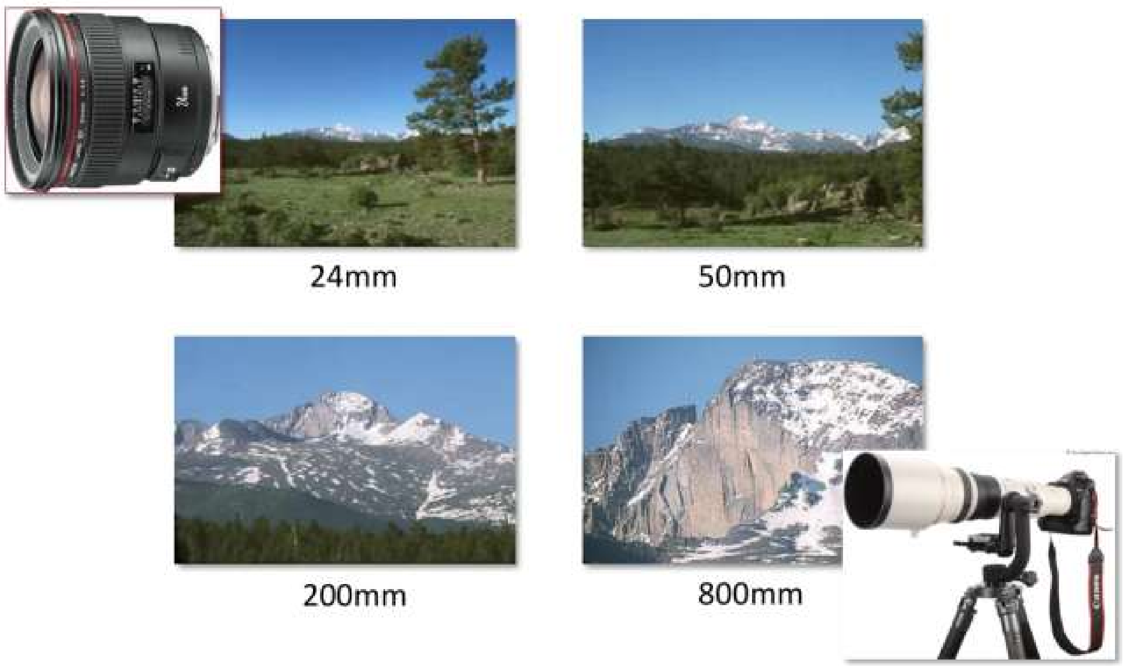


Qué pasa si cambiamos la distancia focal?

Longitud focal

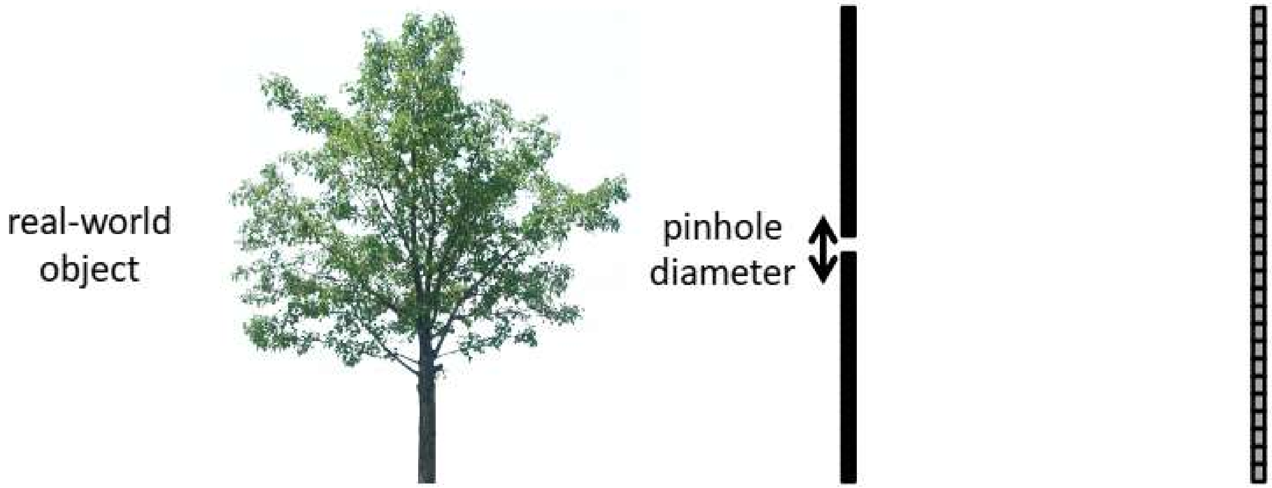
Como un zoom

También relacionado al campo de visión



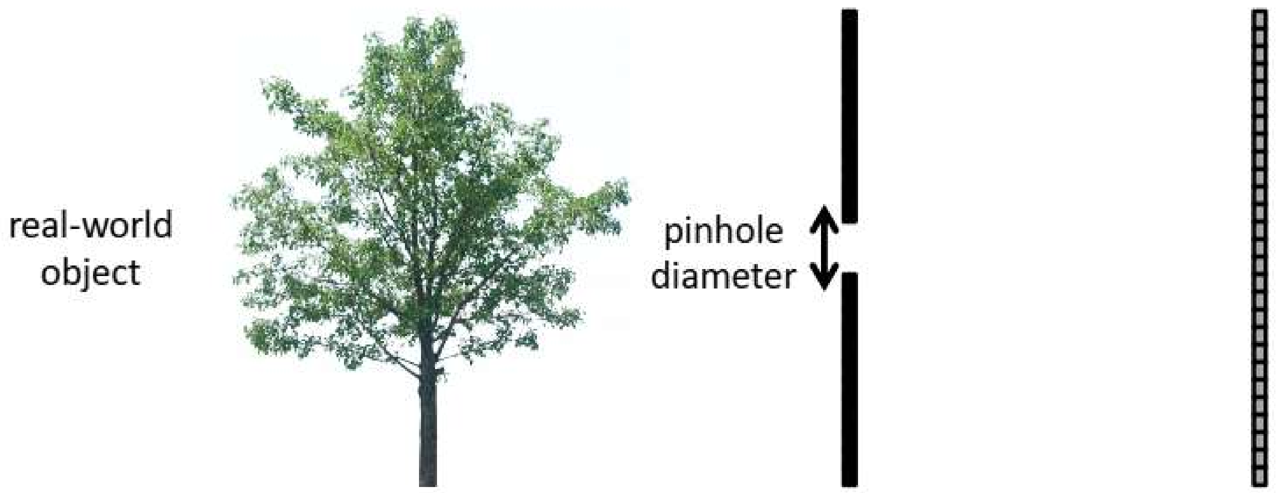
Longitud focal

- Pinhole ideal
- Apertura infinitesimalmente pequeña



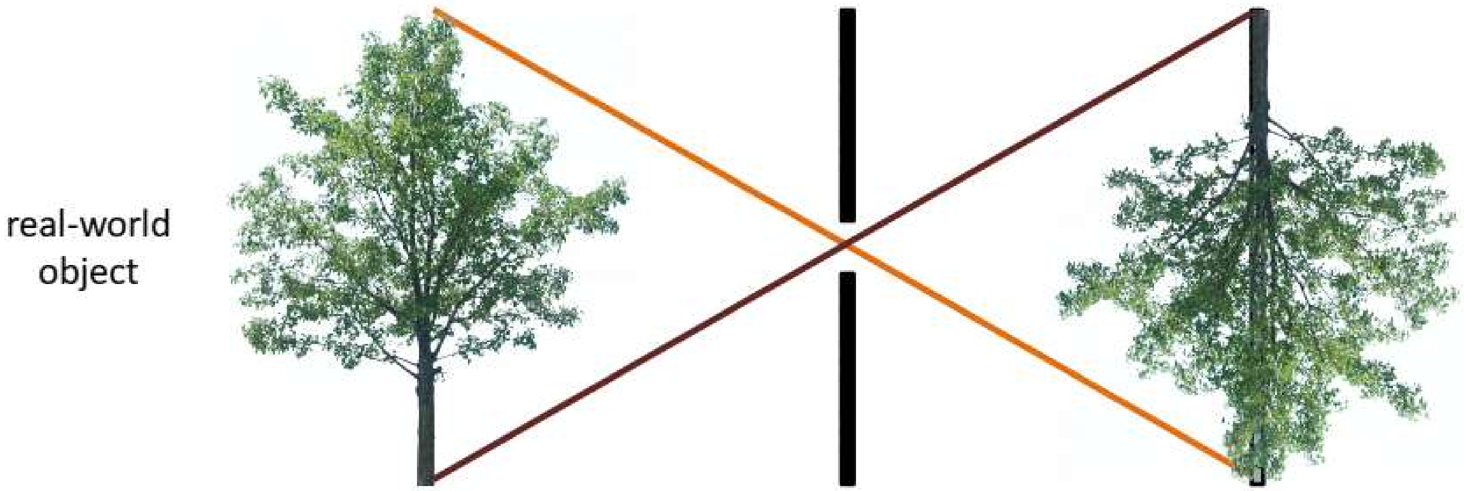
Longitud focal

- Qué pasa si cambiamos el diámetro de la apertura?



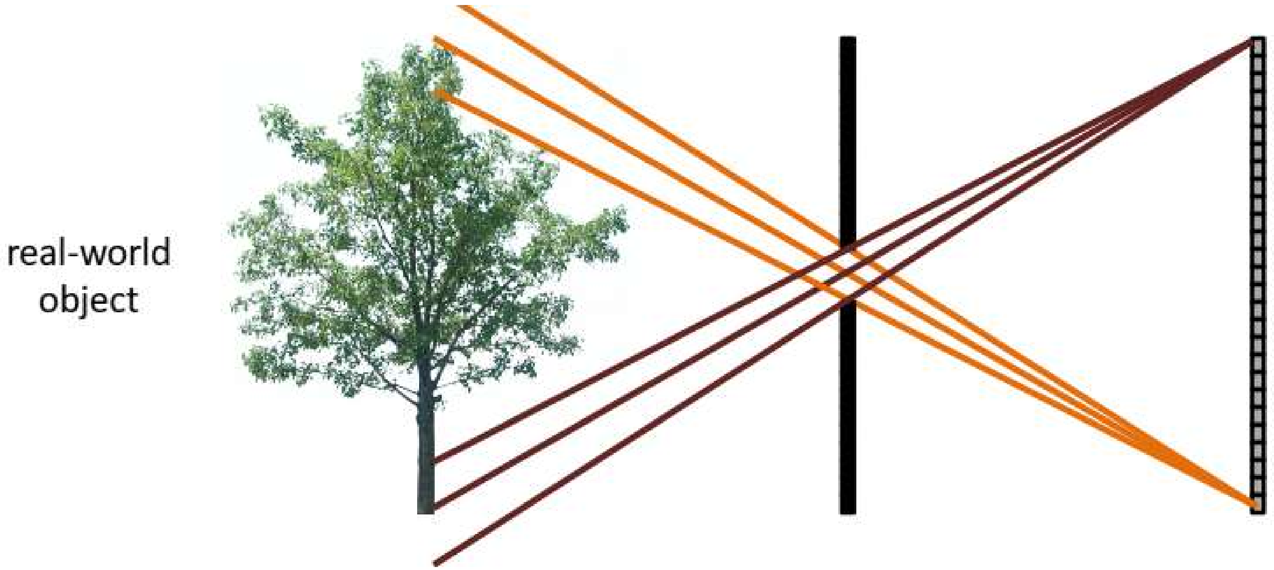
Longitud focal

- Qué pasa si cambiamos el diámetro de la apertura?



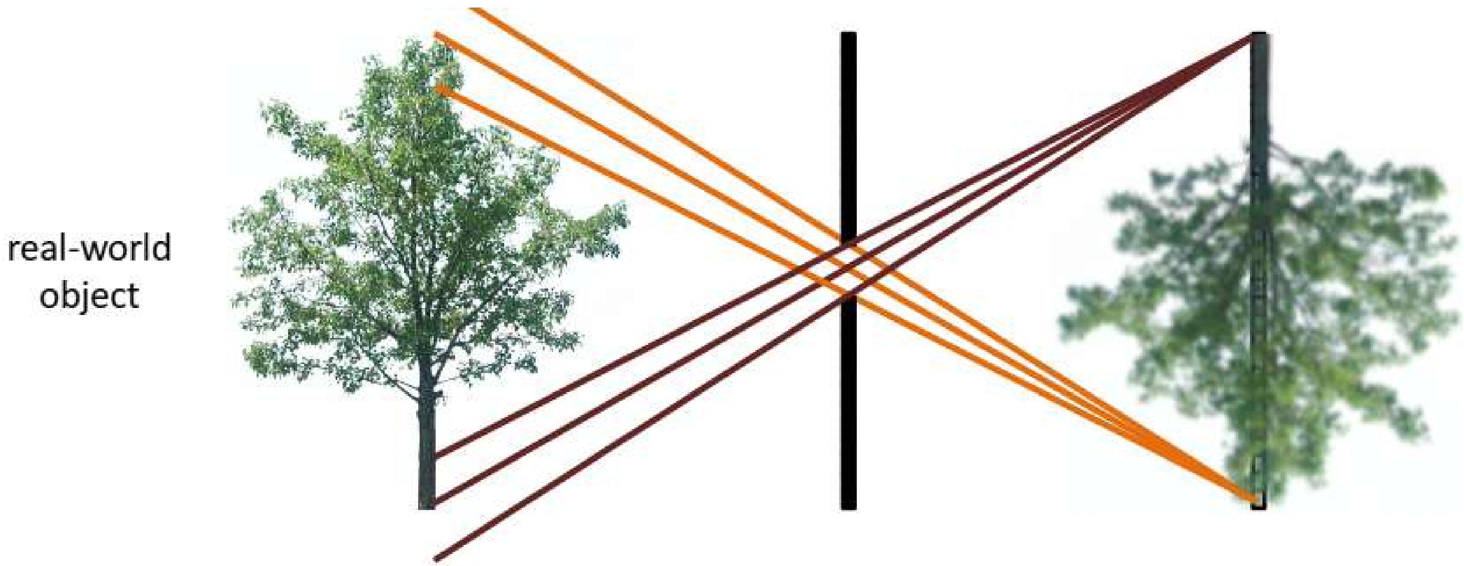
Longitud focal

- Qué pasa si cambiamos el diámetro de la apertura?



Longitud focal

- Qué pasa si cambiamos el diámetro de la apertura?



Proyección

- Imágenes son proyecciones 2D de la escena
- Imágenes capturan dos tipos de información:
  - Geométrica: posiciones, puntos, líneas, etc
  - Fotométrica: intensidad, colores
- Relaciones 2D-3D complejas
- Modelos de cámara aproximan estas relaciones

Proyección

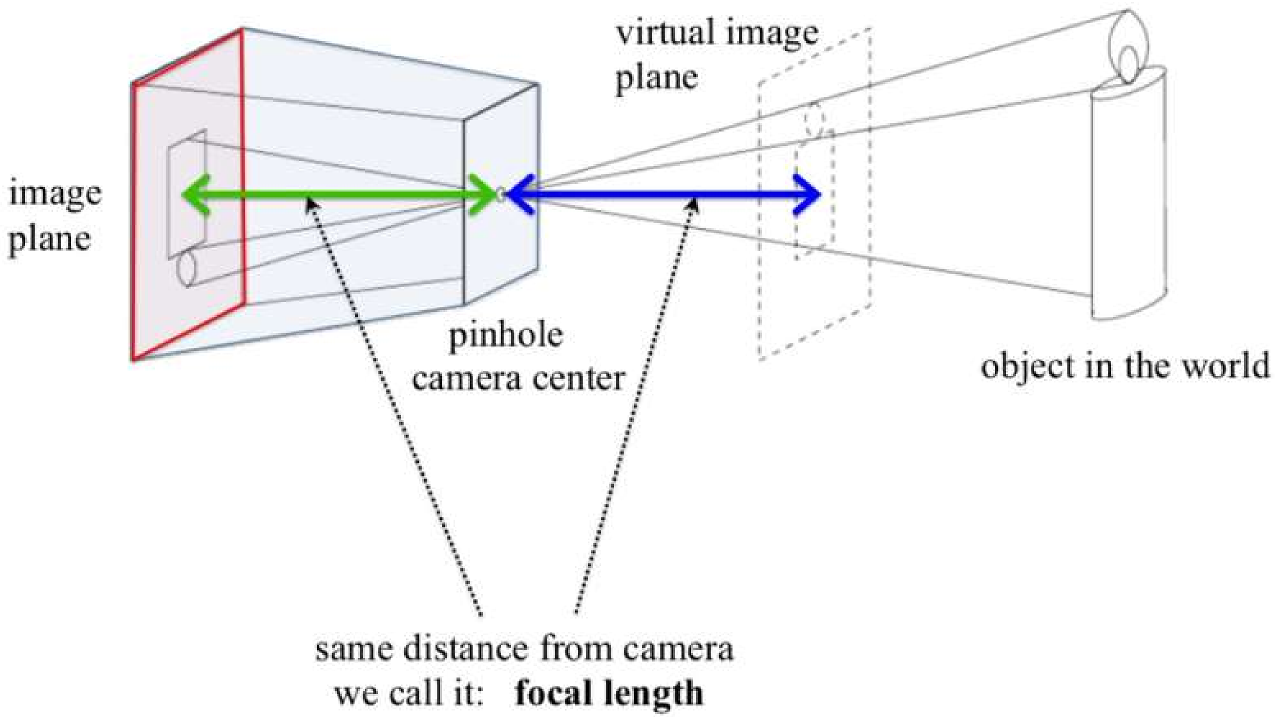
Proyección

Proyección 3D a 2D

- Cómo se proyectan las primitivas 3D a un plano de imagen?
- Con una proyección lineal 3D a 2D
- Diferentes tipos
  - Perspectiva
  - Ortográfica

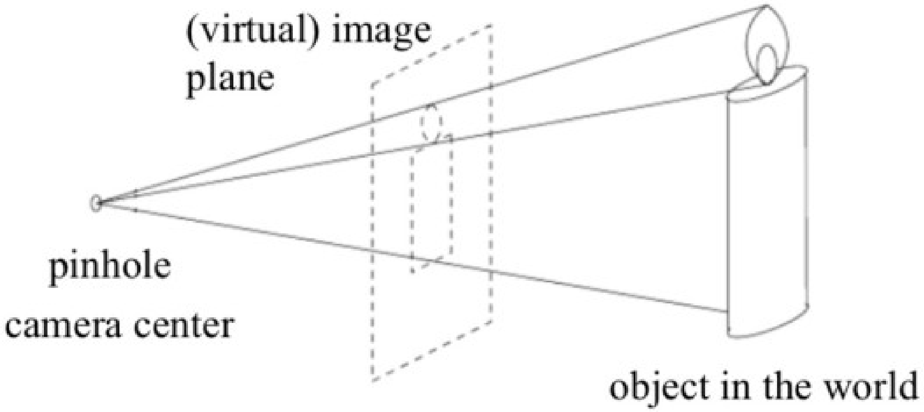
Modelando la proyección

- Usaremos el modelo pinhole como aproximación



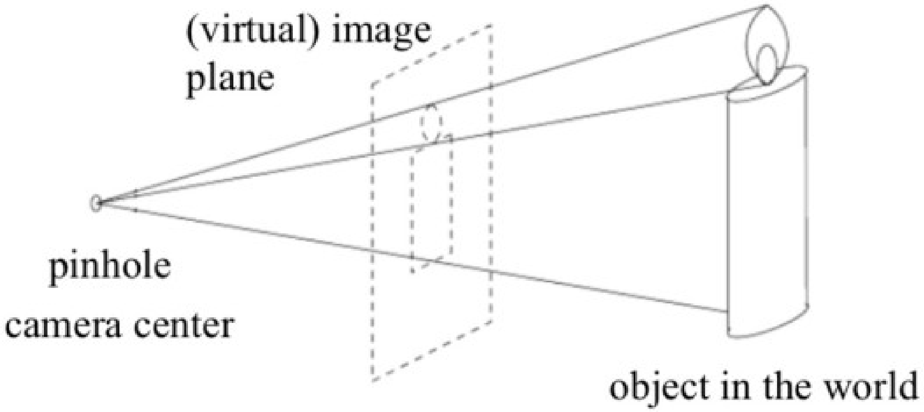
Modelando la proyección

- Usaremos el modelo pinhole como aproximación



- Más fácil trabajar con un plano de imagen virtual
- Si conozco un punto en 3D, puedo computar en qué píxel se proyecta?

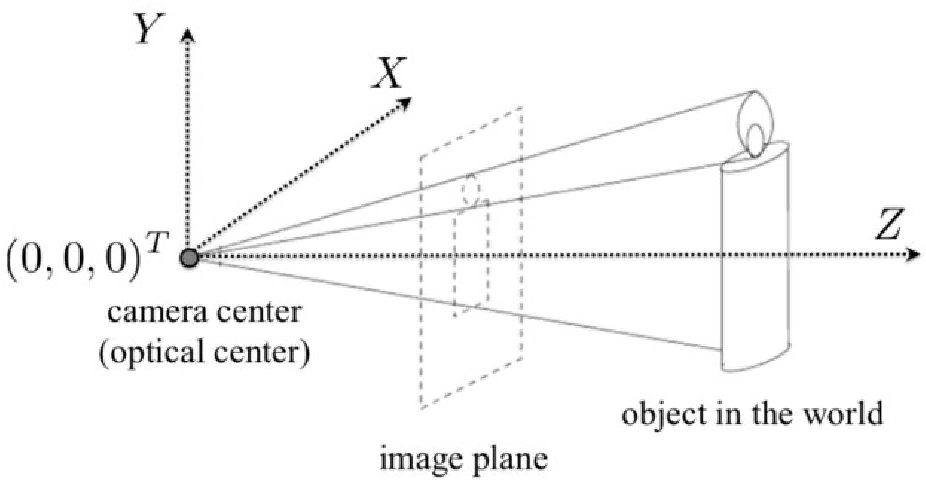
Modelando la proyección



- Necesitamos algo de matemática para derivar
- Necesitamos un sistema coordenado

Modelando la proyección

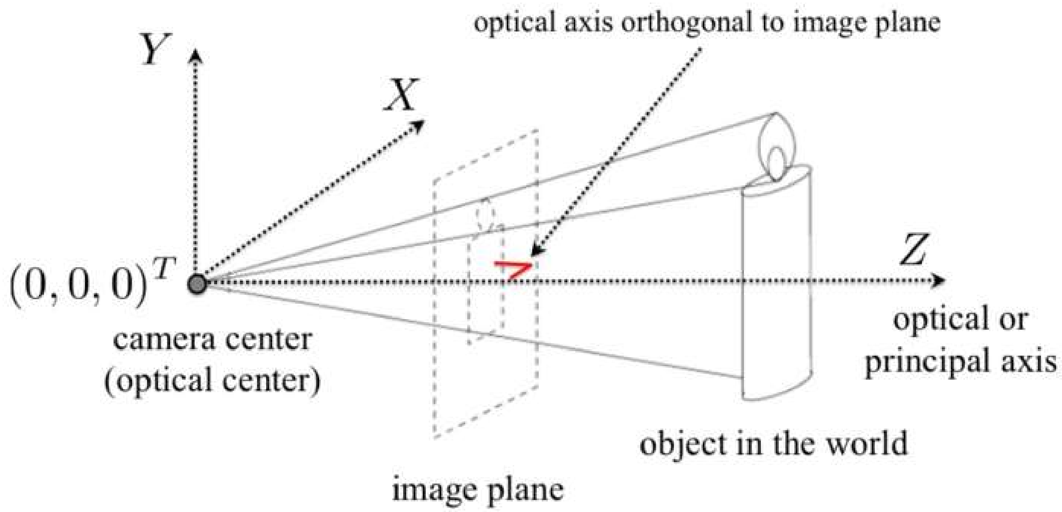
Sistema coordenado de cámara 3D



- Usamor un sistema relativo a la cámara. Centro de cámara está en (0,0,0)

Modelando la proyección

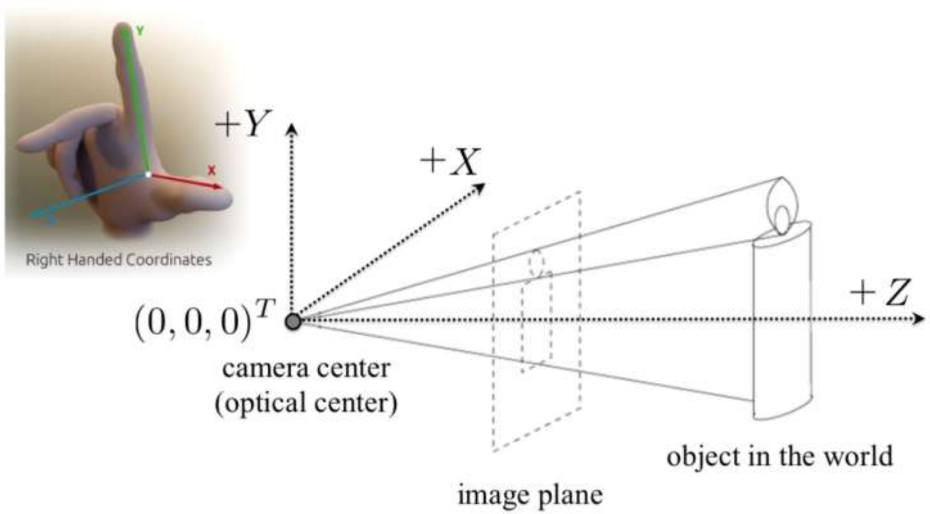
Sistema coordenado de cámara 3D



- El eje Z es llamado eje óptico o principal. Es ortogonal al plano de imagen. Ejes X Y son paralelos a la imagen.

Modelando la proyección

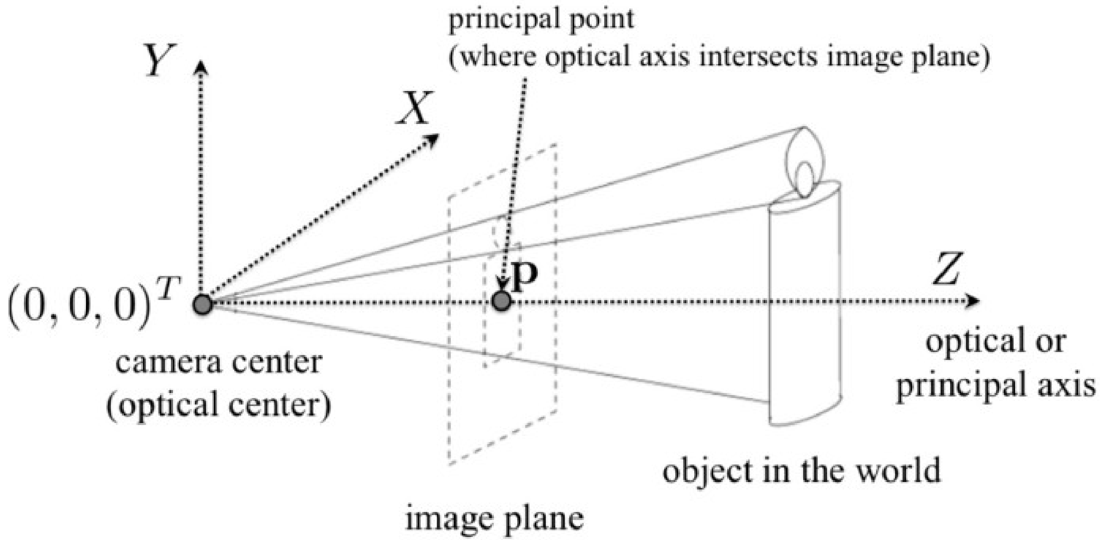
Sistema coordenado de cámara 3D



- Usaremos un sistema de mano derecha

Modelando la proyección

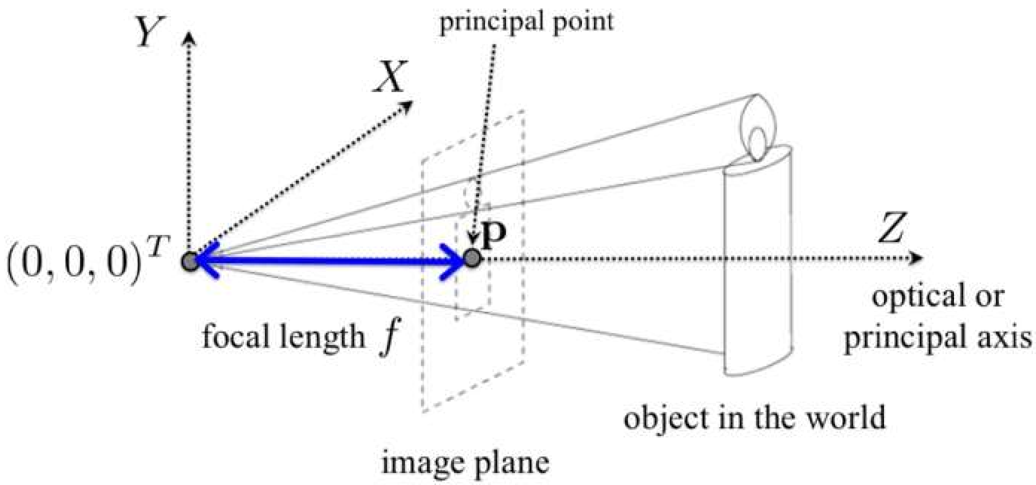
Sistema coordenado de cámara 3D



- El eje óptico intersecta la imagen en un punto $\boldsymbol{p}$. A este punto le llamamos **punto principal**

Modelando la proyección

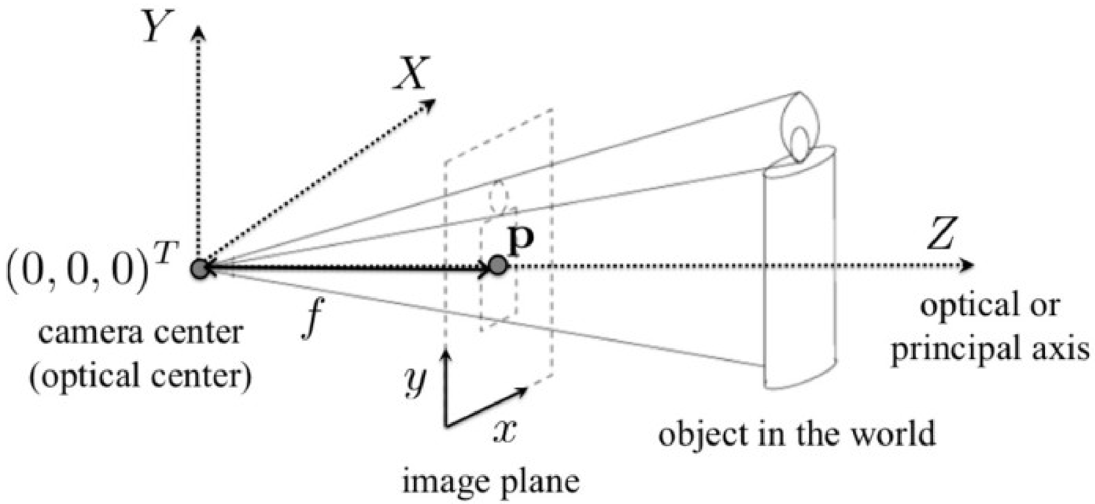
Sistema coordenado de cámara 3D



- La distancia del centro de la cámara al punto principal se llama longitud focal. Lo denotaremos como f

Modelando la proyección

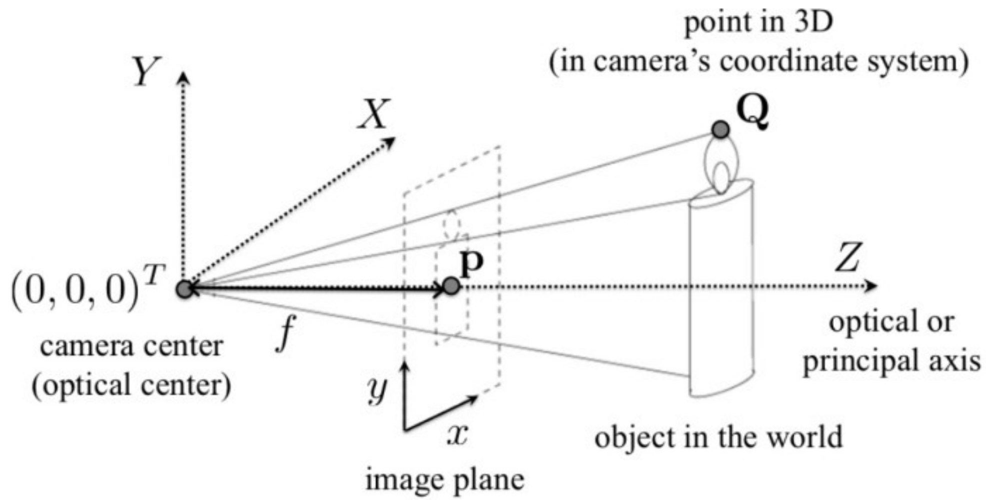
Sistema coordenado de cámara 3D



- Denotaremos los ejes de imagen con x, y. Lo llamaremos sistema coordenado de imagen.

Modelando la proyección

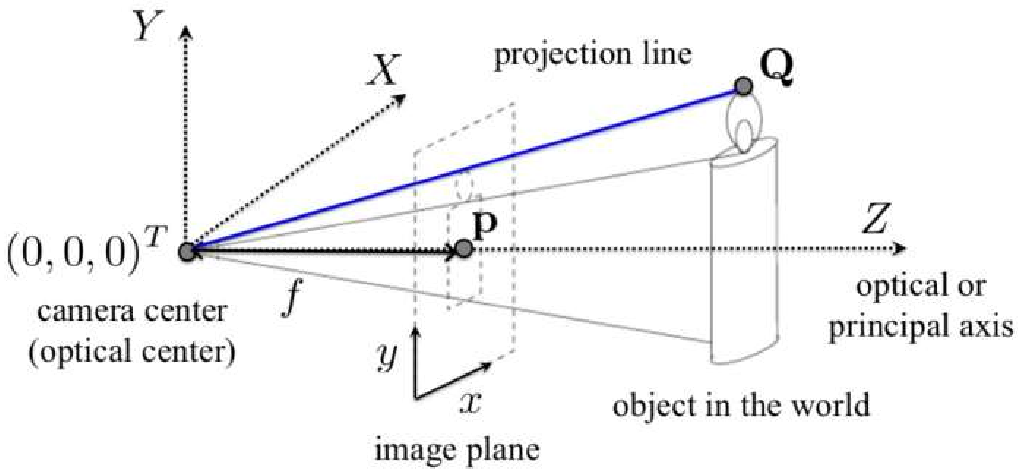
Sistema coordenado de cámara 3D



- Tomemos algún punto Q en 3D. Q vive en el sistema relativo de la cámara

Modelando la proyección

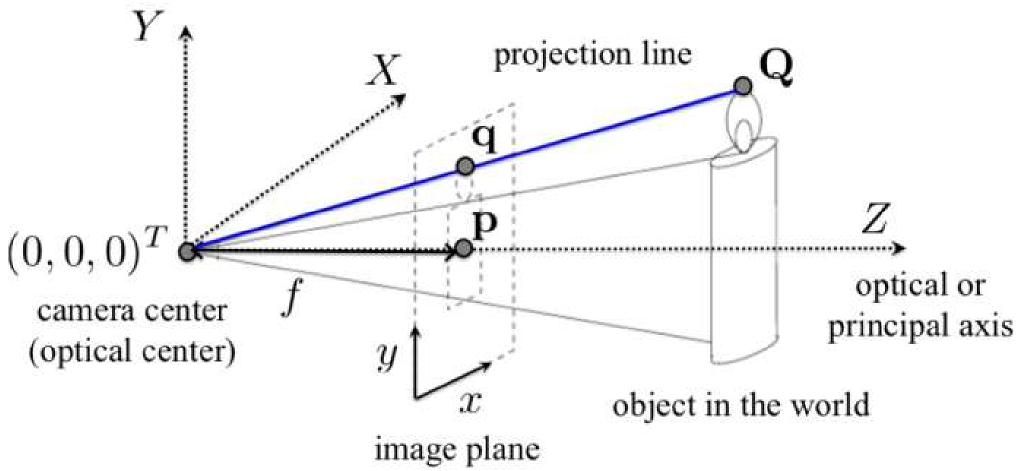
Sistema coordenado de cámara 3D



- La línea desde Q hasta el centro de la cámara se llama **línea de proyección**

Modelando la proyección

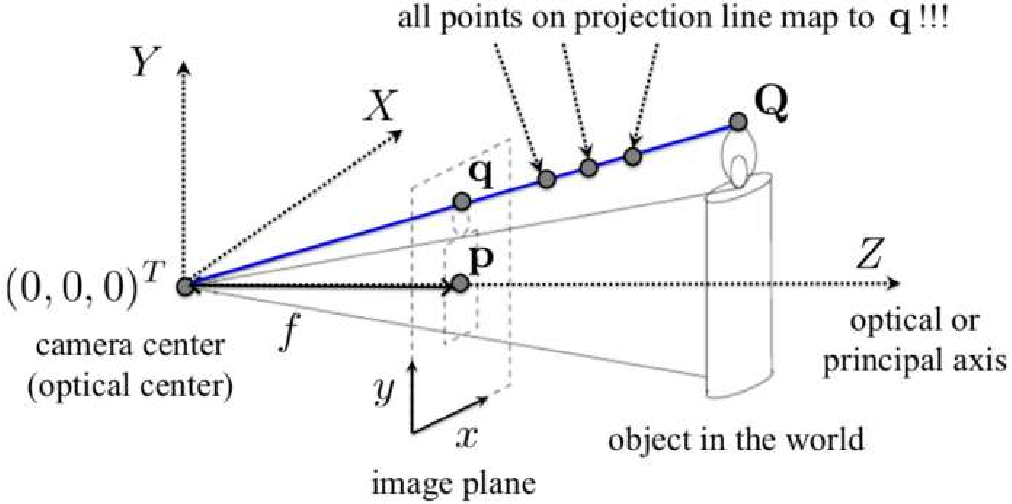
Sistema coordenado de cámara 3D



- La línea de proyección intersecta el plano de la imagen en el punto q. Este es el punto que vemos en la imagen.

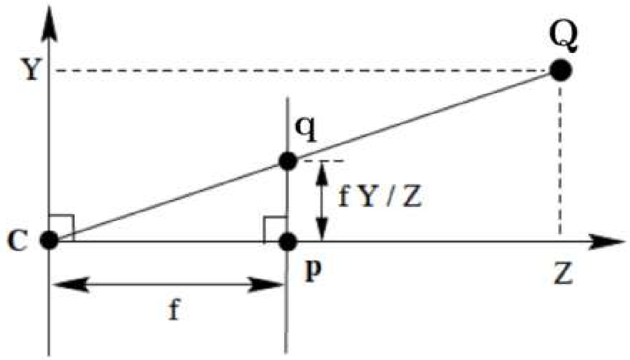
Modelando la proyección

Sistema coordenado de cámara 3D



- Todos los puntos en la línea de proyección de Q se proyectan en $\boldsymbol{q}$.
- Ambigüedad: es imposible conocer la distancia de un punto 3D desde la cámara.

Un poco de matemática

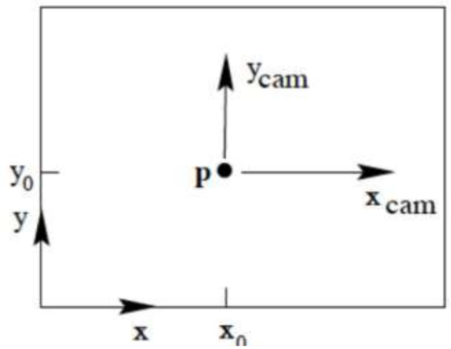


Ecuaciones de proyección

- Usando triángulos similares

$$\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \left(\frac{f \cdot X}{Z}, \frac{f \cdot Y}{Z}, f\right)^T$$

Un poco de matemática



Ecuaciones de proyección

- Usando triángulos similares
- Relativo al punto principal p

$$\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \left(\frac{f \cdot X}{Z}, \frac{f \cdot Y}{Z}, f\right)^T$$

$$\mathbf{q} = (X, Y, Z)^T \quad \rightarrow \quad \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y, f\right)^T$$

Un poco de matemática

Ecuaciones de proyección

- Usando triángulos similares $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \left(\frac{f \cdot X}{Z}, \frac{f \cdot Y}{Z}, f\right)^T$
- Relativo al punto principal p

$$\mathbf{q} = (X, Y, Z)^T \quad \rightarrow \quad \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y, f\right)^T$$

- La proyección se obtiene descartando la última coordenada

$$\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y\right)^T$$

- Pero esta NO es una transformación lineal

Un poco de matemática

Usaremos coordenadas homogéneas

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w) \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

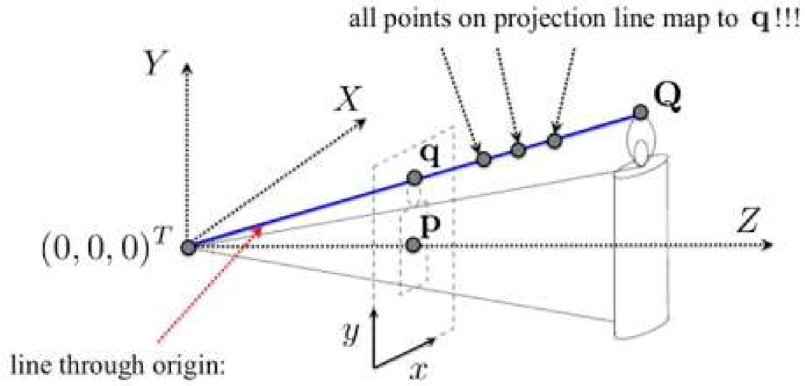
Un poco de matemática

Usaremos coordenadas homogéneas

- En coordenadas homogéneas, el escalamiento no afecta

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda \cdot \begin{bmatrix} X_Q \\ Y_Q \\ Z_Q \end{bmatrix} \quad \rightarrow \quad \mathbf{q}$$



Un poco de matemática

Coordenadas homogéneas son útiles en general

- Ecuación de una línea en 2D: $ax + by + c = 0$
- Se puede representar como $\boldsymbol{I} = (a, b, c)^T$ y $\boldsymbol{x} = (x, y, 1)$
- Línea es $I^T . x = 0$
- Así que si tengo una línea y un punto , rápidamente puedo saber si el punto cae en la línea, chequeando que el producto punto de arriba sea cero.

Proyección perspectiva

- Actualmente tenemos

$$\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \begin{bmatrix} \frac{f \cdot X}{Z} + p_x \\ \frac{f \cdot Y}{Z} + p_y \end{bmatrix}$$

- Lo escribimos como coordenadas homogéneas

$$\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \begin{bmatrix} \frac{f \cdot X}{Z} + p_x \\ \frac{f \cdot Y}{Z} + p_y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f \cdot X + Z \cdot p_x \\ f \cdot Y + Z \cdot p_y \\ Z \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} f \cdot X + Z \cdot p_x \\ f \cdot Y + Z \cdot p_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Intrínsecos de la cámara

$$\mathbf{Q} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} f \cdot X + Z \cdot p_x \\ f \cdot Y + Z \cdot p_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- Lo escribimos

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Llamada matriz de calibración de cámara o matriz de parámetros intrínsecos

Intrínsecos de la cámara

- Matriz de calibración

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

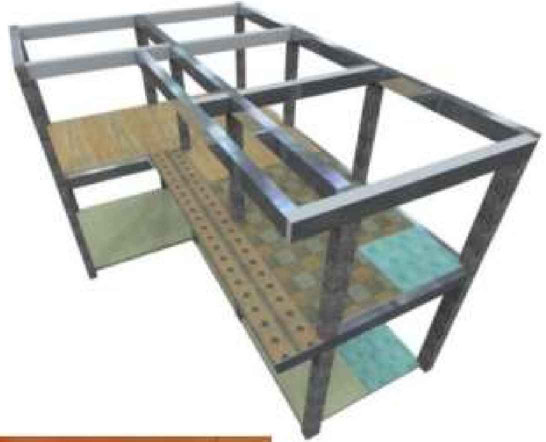
- Los píxeles podrían no ser cuadrados

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Y podría existir un ángulo entre los ejes de la imagen

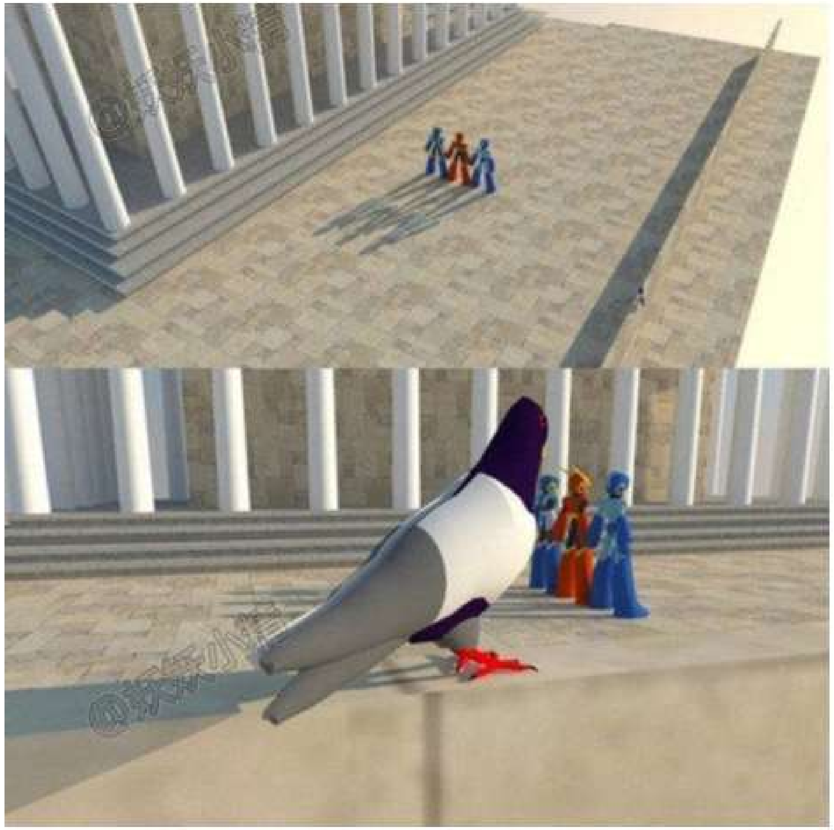
$$K = \begin{bmatrix} f_x & -f_x \cot\theta & p_x \\ 0 & f_y / \sin\theta & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Proyección perspectiva

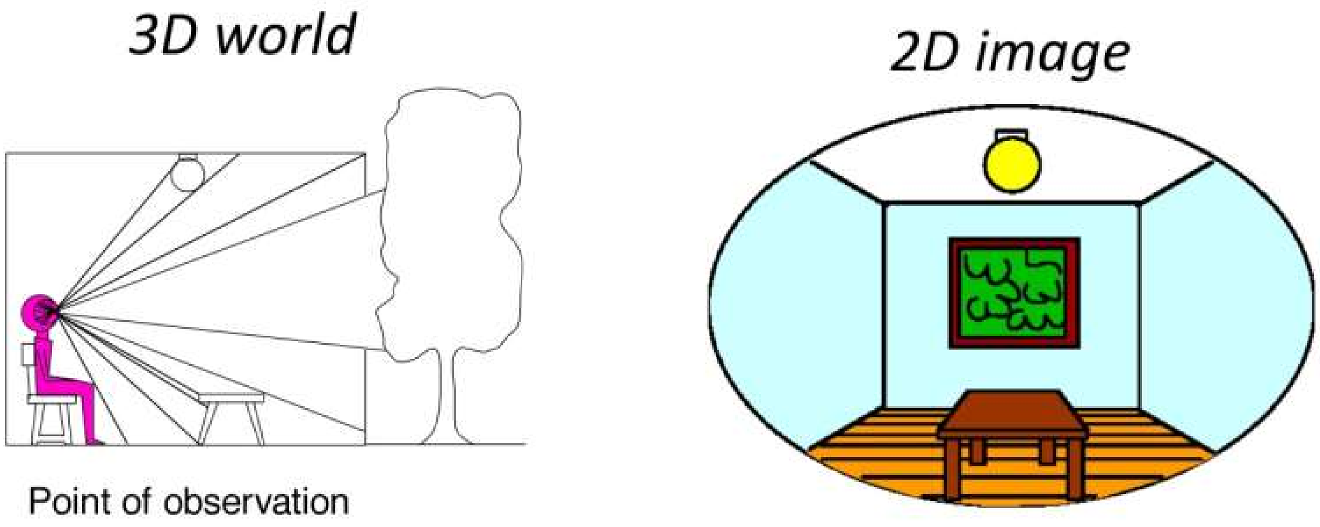


Proyección perspectiva

Proyección perspectiva



Reducción de dimensionalidad



- Se pierden ángulos y distancias

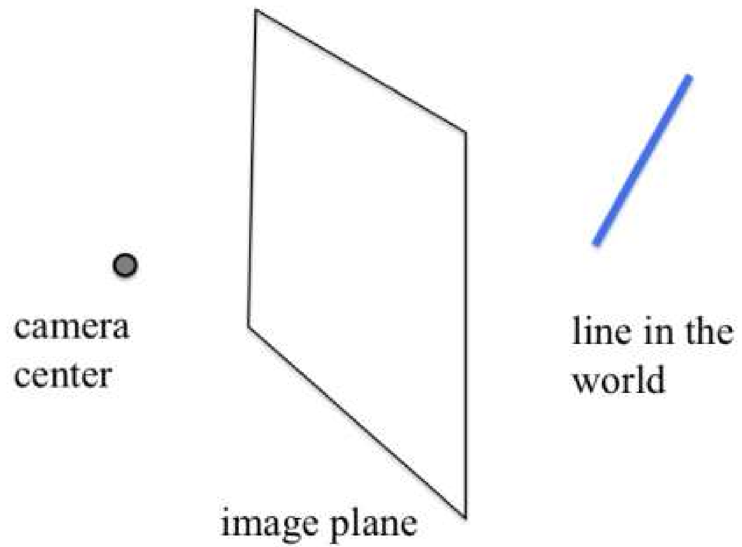
Propiedades de la proyección

- Muchos a uno: cualquier punto en la línea de proyección se mapea igual
- Puntos a puntos
- Líneas a líneas

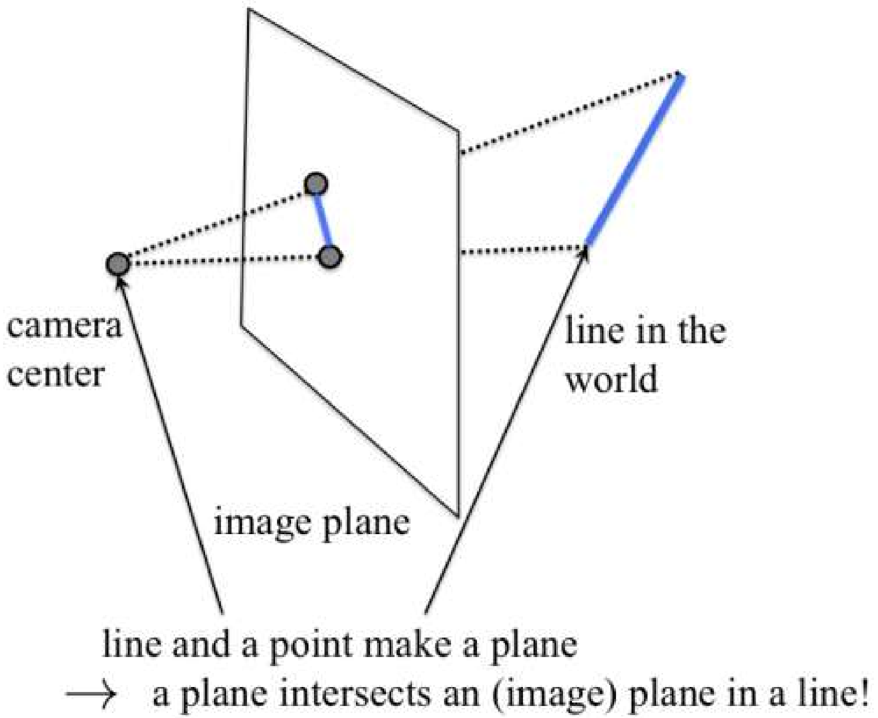
Propiedades de la proyección

- Muchos a uno: cualquier punto en la línea de proyección se mapea igual
- Puntos a puntos
- Líneas a líneas

Propiedades de la proyección



Propiedades de la proyección



Propiedades de la proyección

- Línea que se proyecta en el punto principal → punto

Propiedades de la proyección

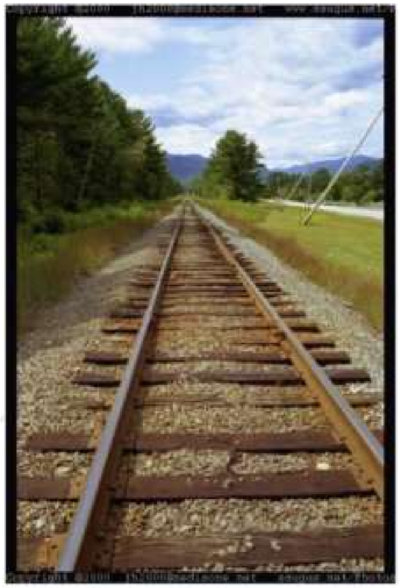
- Planos a planos

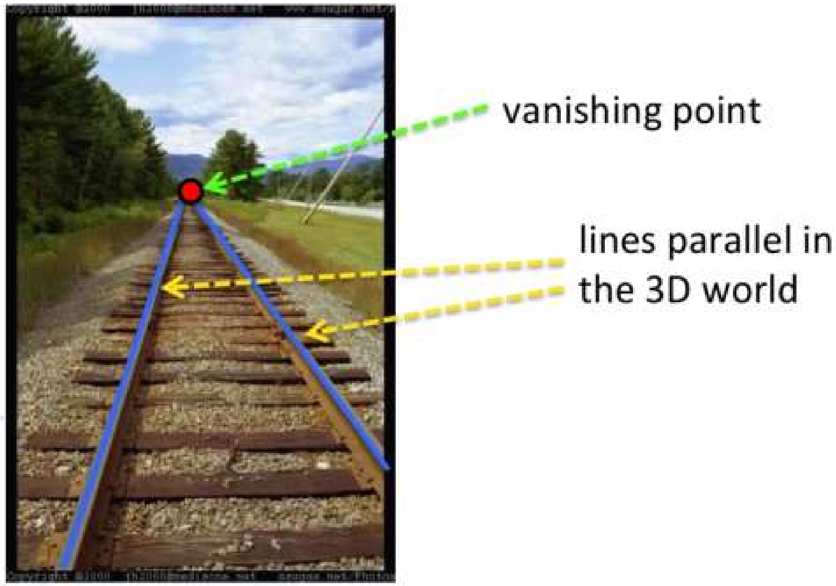
Propiedades de la proyección

- Planos a planos
- Plano a través del punto principal que es ortogonal al plano de imagen se proyecta en línea

Propiedades de la proyección

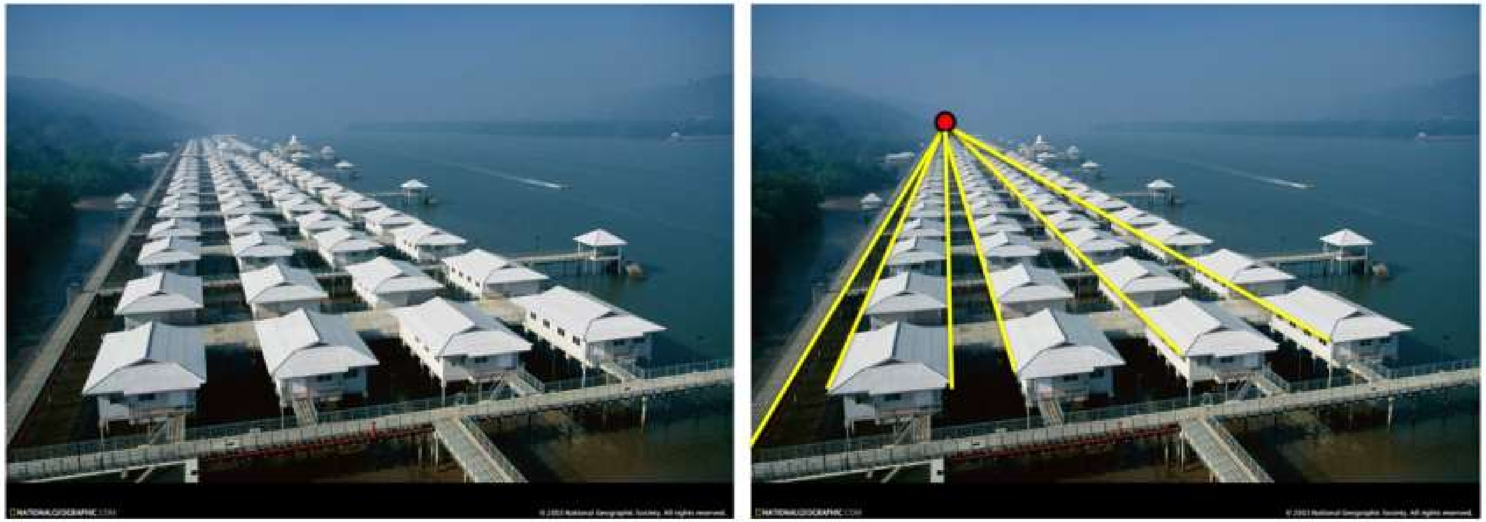
- Líneas paralelas convergen en un punto de desvanecimiento





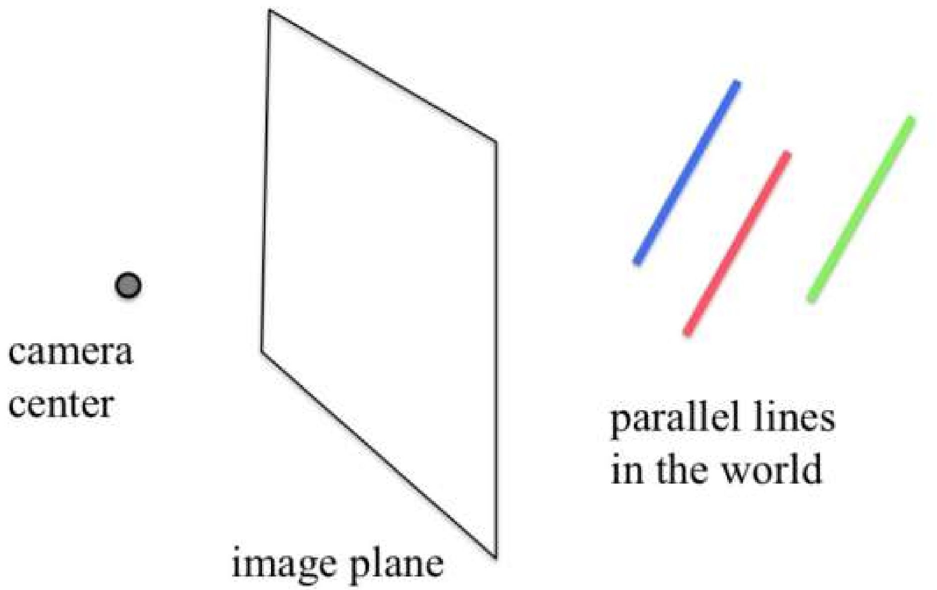
Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento



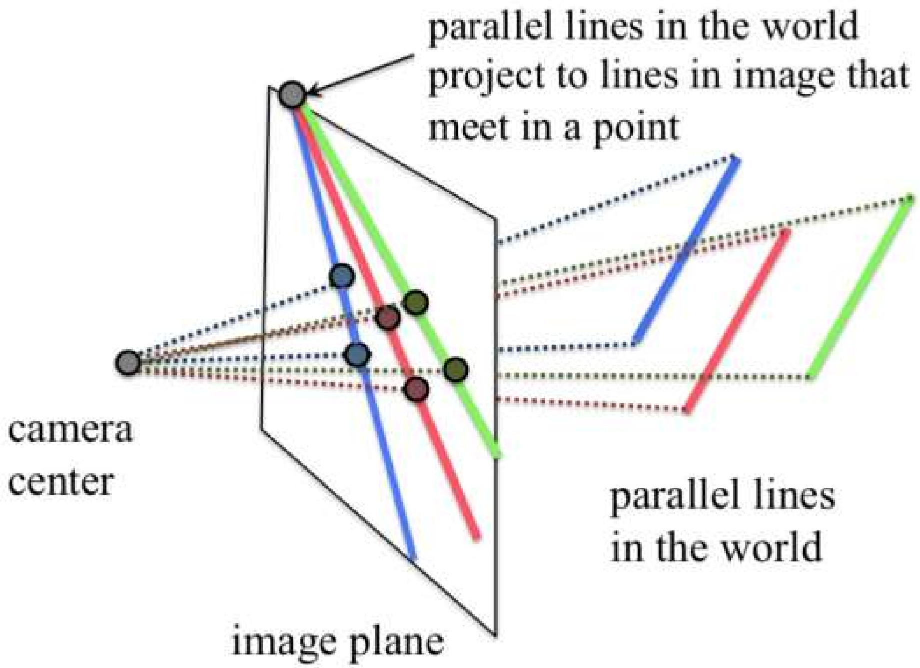
Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento



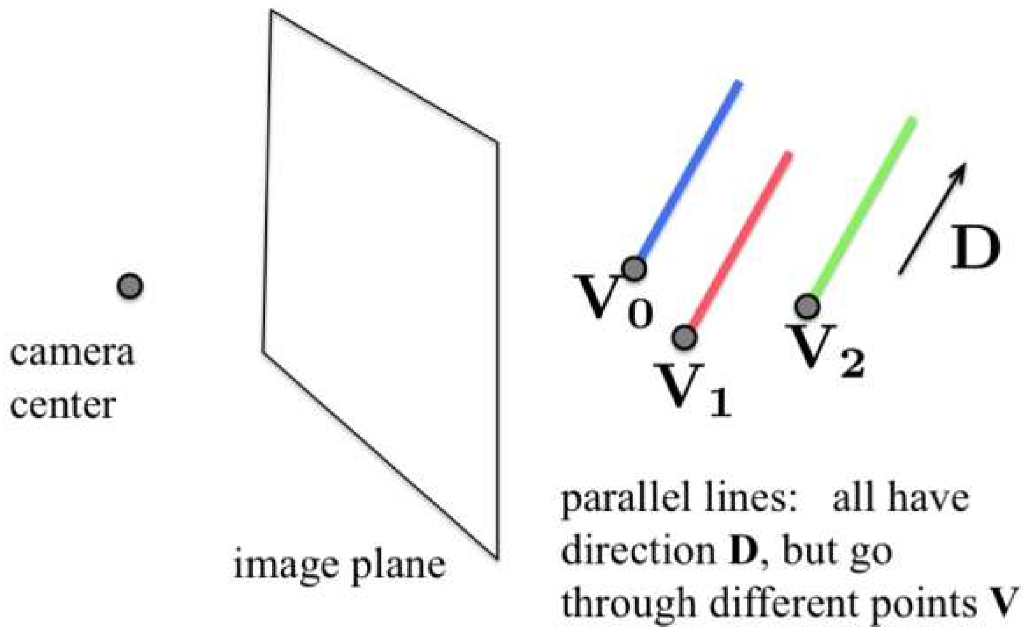
Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento



Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento



Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento
- Líneas que pasan por V y tienen dirección D: $X = V + tD$

Propiedades de la proyección

- Todas las líneas con la misma dirección 3D convergen al mismo punto de desvanecimiento
- Líneas que pasan por V y tienen dirección D: $X = V + tD$
- Proyectamos

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = K\mathbf{X} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_x + tD_x \\ V_y + tD_y \\ V_z + tD_z \end{bmatrix} = \begin{bmatrix} fV_x + ftD_x + p_x V_z + tp_x D_z \\ fV_y + ftD_y + p_y V_z + tp_y D_z \\ V_z + tD_z \end{bmatrix}$$

- Computamos x, y cuando $t \to \infty$

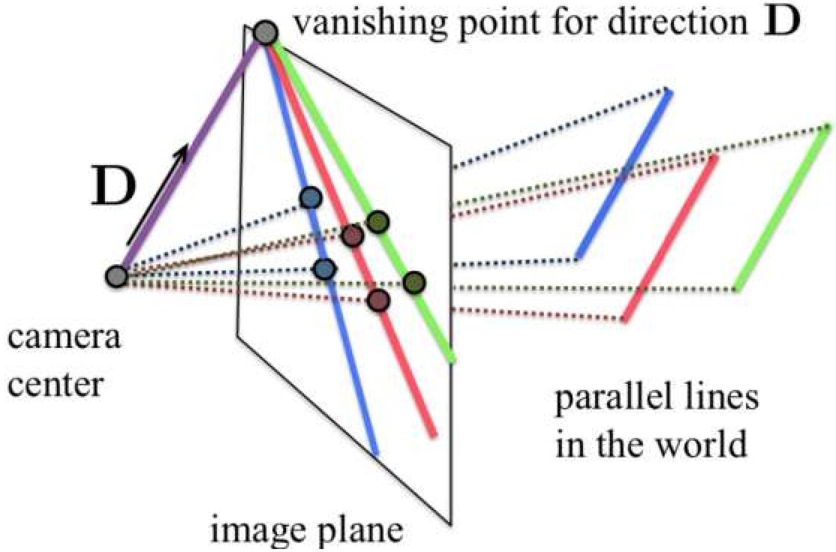
$$x = \lim_{t\to\infty} \frac{fV_x + ftD_x + p_x V_z + tp_x D_z}{V_z + tD_z} = \frac{fD_x + p_x D_z}{D_z}$$

$$y = \lim_{t\to\infty} \frac{fV_y + ftD_y + p_y V_z + tp_y D_z}{V_z + tD_z} = \frac{fD_y + p_y D_z}{D_z}$$

No depende de V

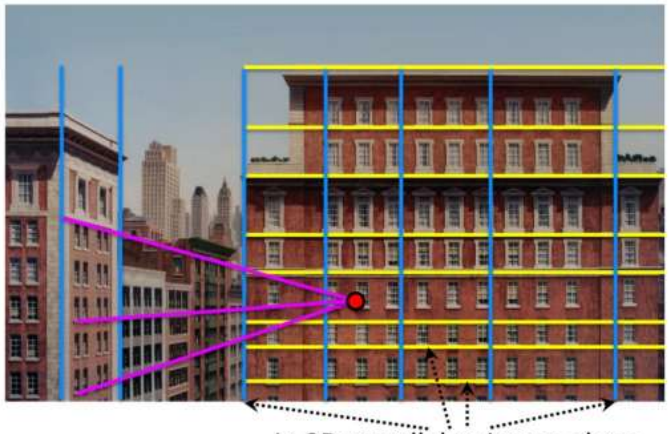
Propiedades de la proyección

- Hay una forma más fácil: trasladar línea D al centro de cámara. Esta línea intersecta el plano de la imagen en el punto de desvanecimiento.



Propiedades de la proyección

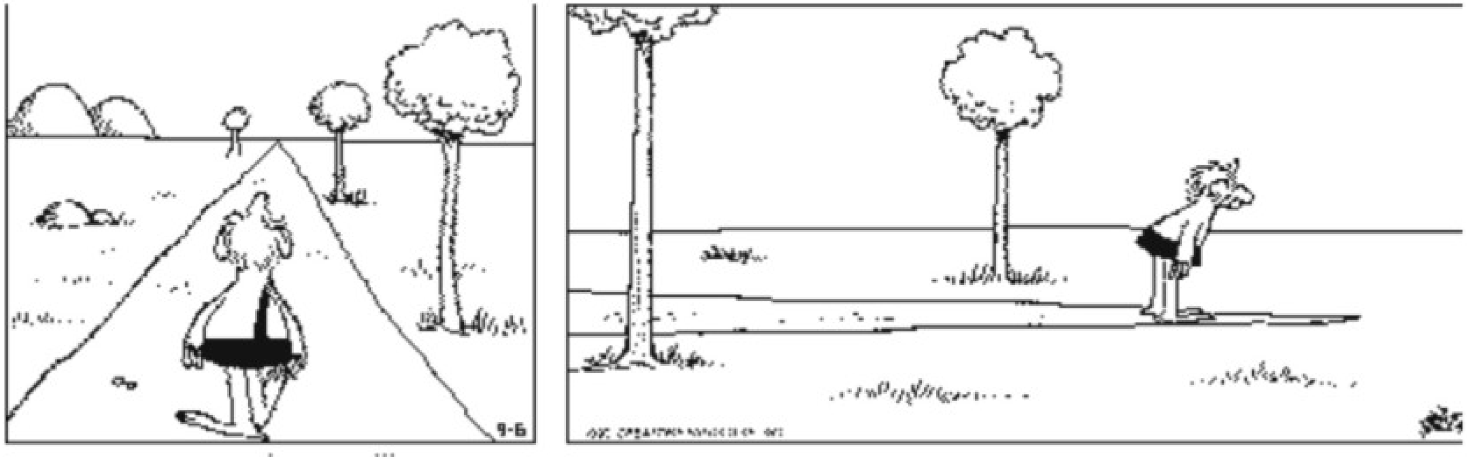
- Líneas paralelas convergen en el punto de desvanecimiento
- Cada difección diferente en el mundo tiene su propio punto de desavanecimiento
- Líneas paralelas al plano de la imagen permanecen paralelas. “Intersectan en el infinito”



In 3D: parallel to image plane
In 2D: parallel (no VP)

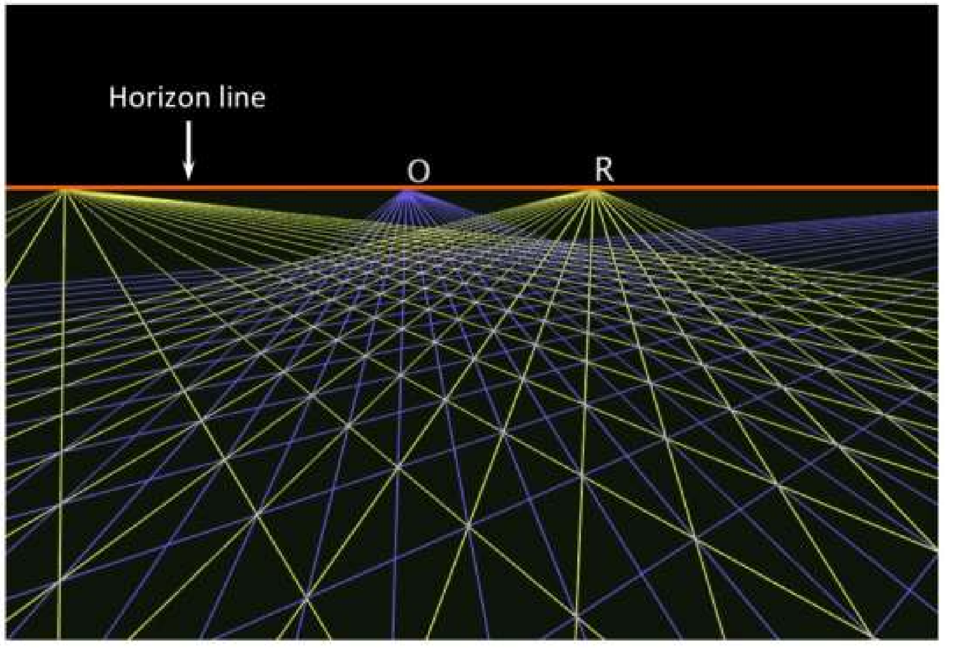
Propiedades de la proyección

- Lineas paralelas convergen en un punto de desvanecimiento
- Pero líneas que se intersectan en 2D no necesariamente son paralelas en 3D



Propiedades de la proyección

- Lineas paralelas convergen en un punto de desvanecimiento
- Para líneas en el mismo plano 3D, los puntos de desvanecimiento caen en una línea. La llamamos línea de desvanecimiento



Propiedades de la proyección

- Planos paralelos en 3D tienen la misma línea de horizonte

Propiedades de la proyección

- Podemos decir qué tan arriba fue tomada esta forografía?

Propiedades de la proyección

- Podemos decir qué tan arriba fue tomada esta forografía?

Propiedades de la proyección

- Podemos decir qué tan arriba fue tomada esta forografía?

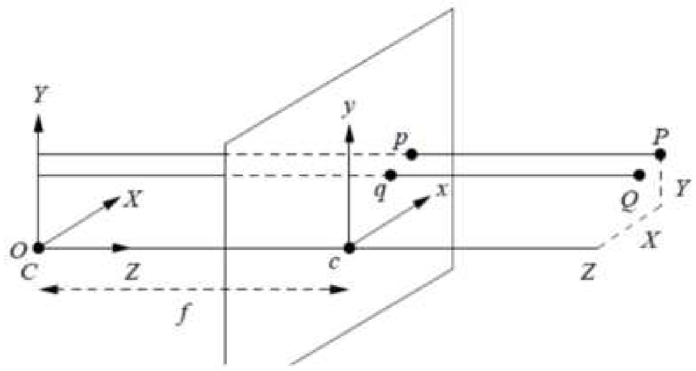
Propiedades de la proyección

- Podemos decir qué tan arriba fue tomada esta forografía?

Propiedades de la proyección

- Solo posible si la cámara es ortogonal al suelo y el suelo es plano

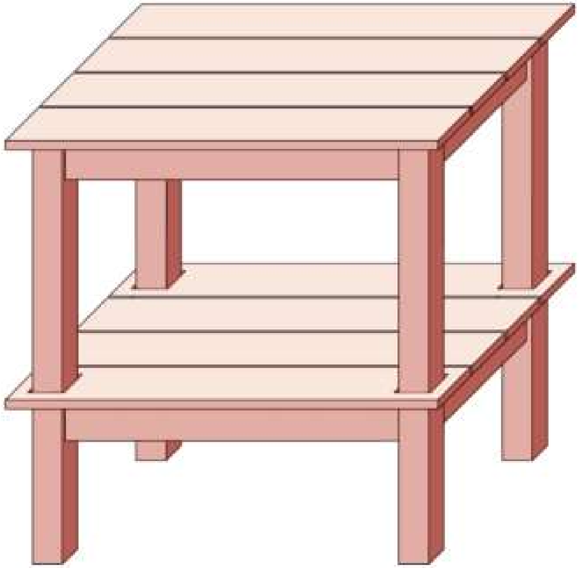
Proyección ortográfica



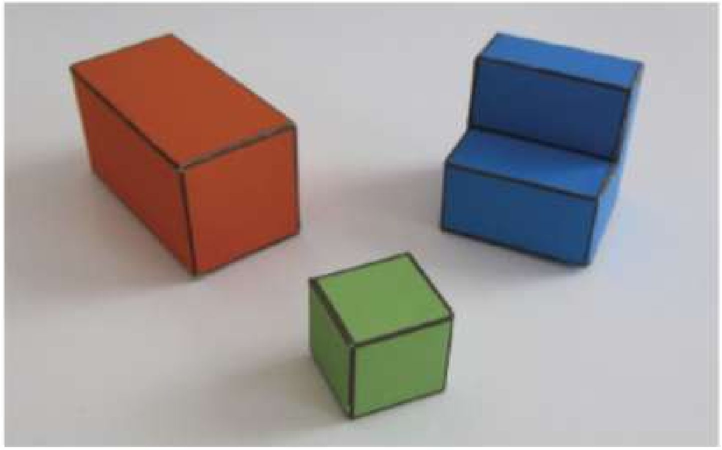
- No requiere división y simplemente se descarta la coordenada Z

$$\mathbf{Q} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \rightarrow \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

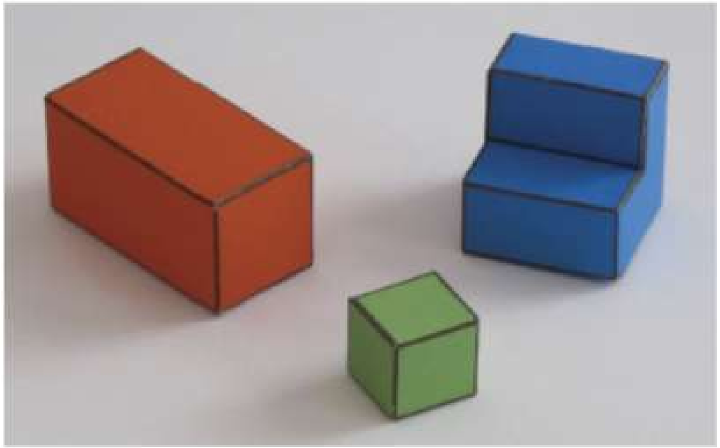
Proyección ortográfica



Proyección ortográfica



Perspective projection



Parallel (orthographic) projection

- Perspectiva: líneas paralelas en 3D no son paralelas en imagen
- Ortográfica: líneas paralelas en 3D son paralelas en imagen

Parámetros de cámara

- Para describir una proyección completamente, necesitamos
  - Describir sus parámetros internos (matriz K)
  - Describir su posición en el mundo. Dos sistemas coordenados
    - Sistema del mundo
    - Sistema de la cámara

Parámetros de cámara

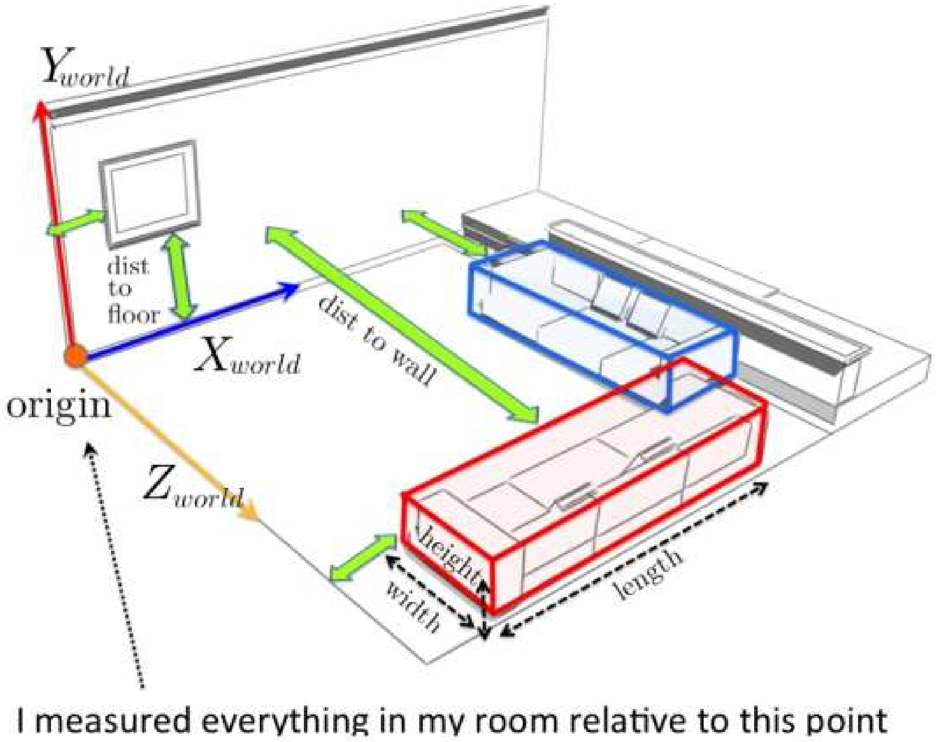
- Porqué dos sistemas coordenados?

Parámetros de cámara

- Porqué dos sistemas coordenados?

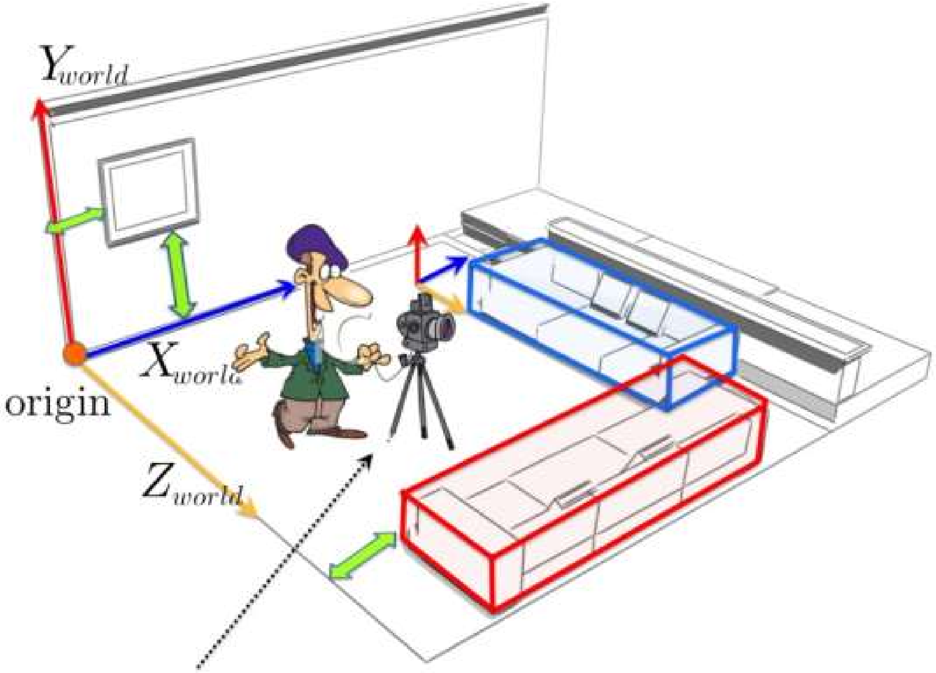
Parámetros de cámara

- Porqué dos sistemas coordenados?



Parámetros de cámara

- Porqué dos sistemas coordenados?

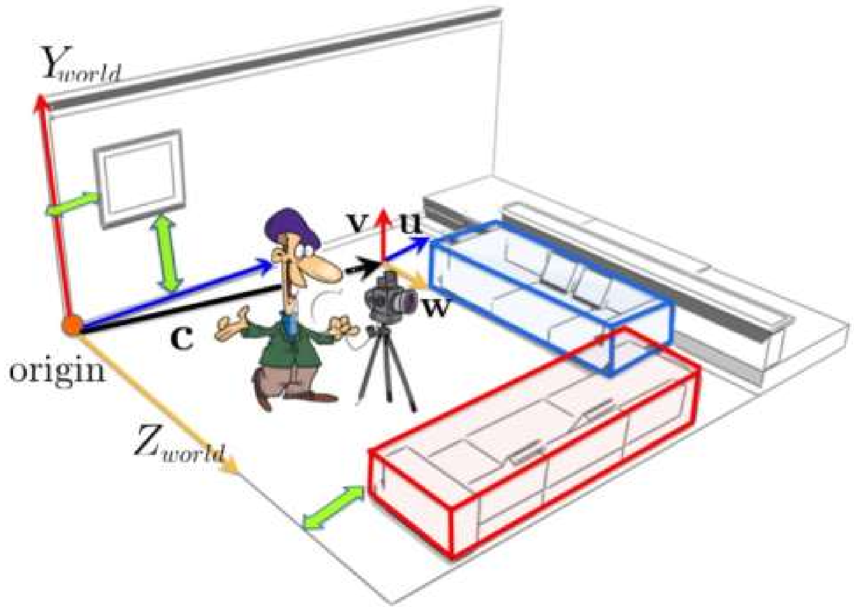


But to project my room to camera, I need to have the room in the camera coordinate system!

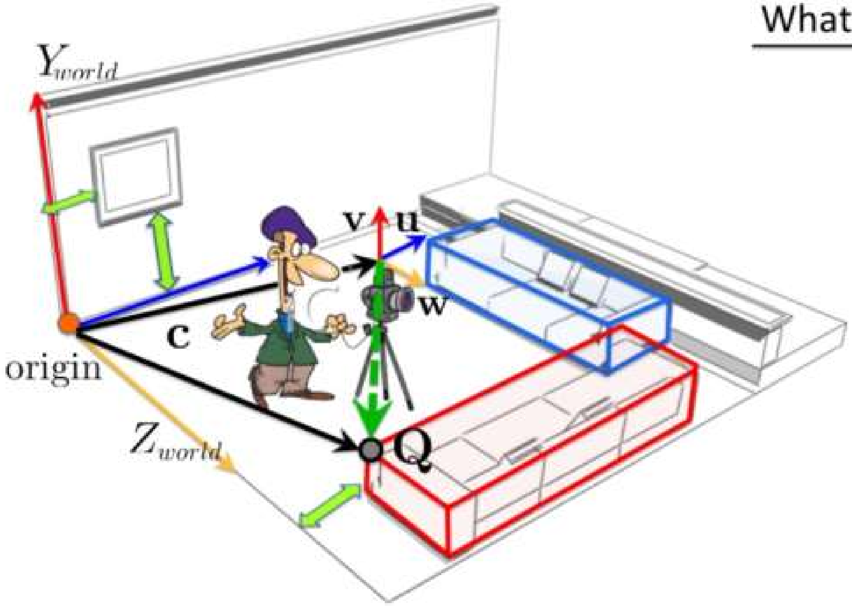
Proyección

- Para proyectar un punto (X, Y, Z) en coordenadas del mundo al plano de la imagen, necesitamos
  - Transformar (X, Y, Z) en coordenadas de cámara
    - Posición de la cámara (en coordenadas del mundo)
    - Orientación de la cámara (en coordenadas del mundo)
  - Proyectar en el plano de imagen
    - Necesitamos los parámetros intrínsecos de las cámaras
  - Todas pueden ser descritas con matrices

Camera Extrinsics

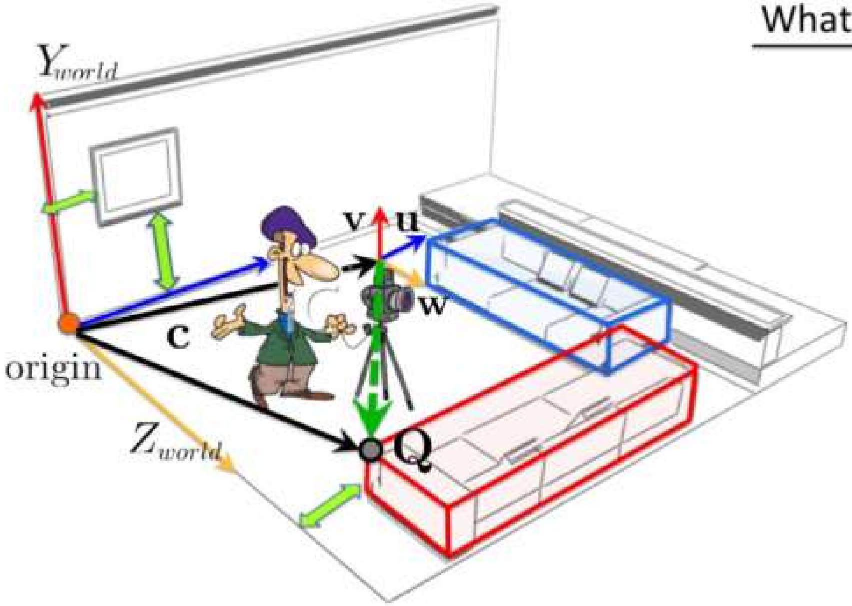


Camera Extrinsics



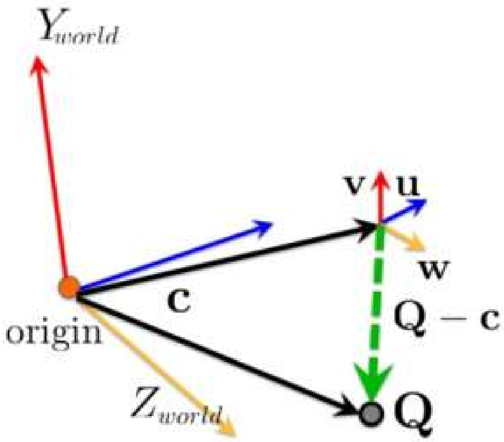
What is Q in camera's coordinate system??

Camera Extrinsics



What is Q in camera's coordinate system??

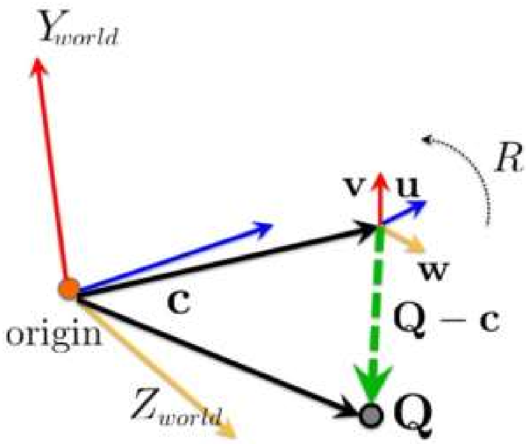
Camera Extrinsics



What is Q in camera's coordinate system??

$\mathbf{Q} - \mathbf{c}$ ... makes **position** relative to camera

Camera Extrinsics

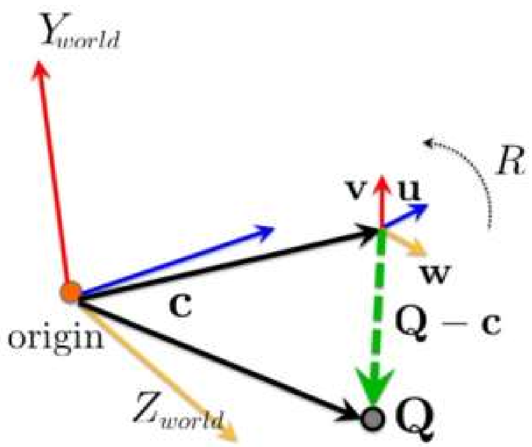


<u>What is Q in camera's coordinate system??</u>

$\mathbf{Q} - \mathbf{c}$ ... makes **position** relative to camera

$R\begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix} = I$ (looking for rotation to canonical orientation)

Camera Extrinsics



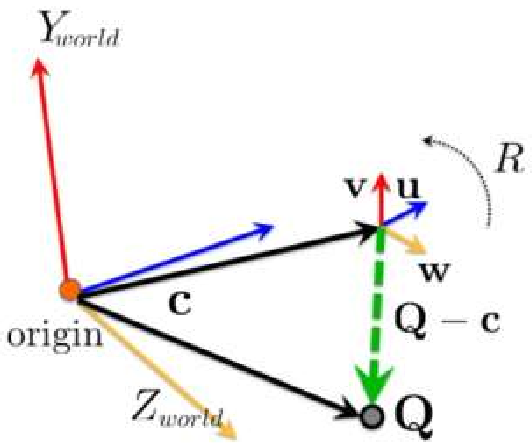
What is Q in camera's coordinate system??

$\mathbf{Q}-\mathbf{c}$ ... makes **position** relative to camera

$R\begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix} = I$ (looking for rotation to canonical orientation)

$R \cdot R^T = I$ (since orientation is orthogonal matrix)

Camera Extrinsics



What is Q in camera's coordinate system??

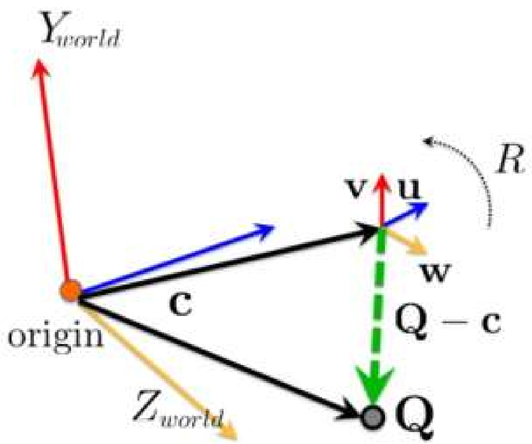
$\mathbf{Q} - \mathbf{c}$ ... makes **position** relative to camera

$R\begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix} = I$ (looking for rotation to canonical orientation)

$R \cdot R^T = I$ (since orientation is orthogonal matrix)

$R = \begin{bmatrix}\mathbf{u}^T & \mathbf{v}^T & \mathbf{w}^T\end{bmatrix}$

Camera Extrinsics



What is Q in camera's coordinate system??

$\mathbf{Q}-\mathbf{c}$ ... makes **position** relative to camera

$R\begin{bmatrix}\mathbf{u} & \mathbf{v} & \mathbf{w}\end{bmatrix} = I$ (looking for rotation to canonical orientation)

$R \cdot R^T = I$ (since orientation is orthogonal matrix)

$R = \begin{bmatrix}\mathbf{u}^T & \mathbf{v}^T & \mathbf{w}^T\end{bmatrix}$

Transformación Final

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \mathbf{c}\right) = \begin{bmatrix} R & -Rc \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

camera coordinates

room (world) coordinates

Ecuaciones de proyección

- Matriz de proyección $\boldsymbol{P}$ modela el efecto acumulativo de todos los parámetros intrínsecos y extrínsecos. Usamos coordenadas homogéneas

$$\mathbf{q} = \begin{bmatrix} ax \\ ay \\ a \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Se puede computar como

$$\mathbf{P} = \underbrace{\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics } K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{T}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}}_{\text{translation}}$$

Ecuaciones de proyección

- La matriz de proyección es definida como

$$\mathbf{P} = \underbrace{\mathbf{K}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\underbrace{\begin{bmatrix} \mathbf{R}_{3\times 3} & 0_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{T}_{3\times 3} \\ 0_{1\times 3} & 1 \end{bmatrix}}_{\text{translation}}}_{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}}$$

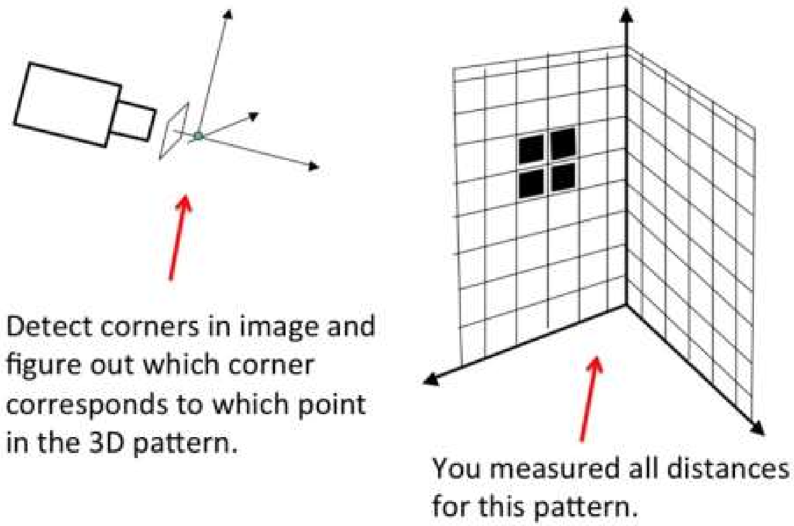
- Más compactamente

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

- Esto es complicado, porque en general, no tenemos P

Calibración de cámaras

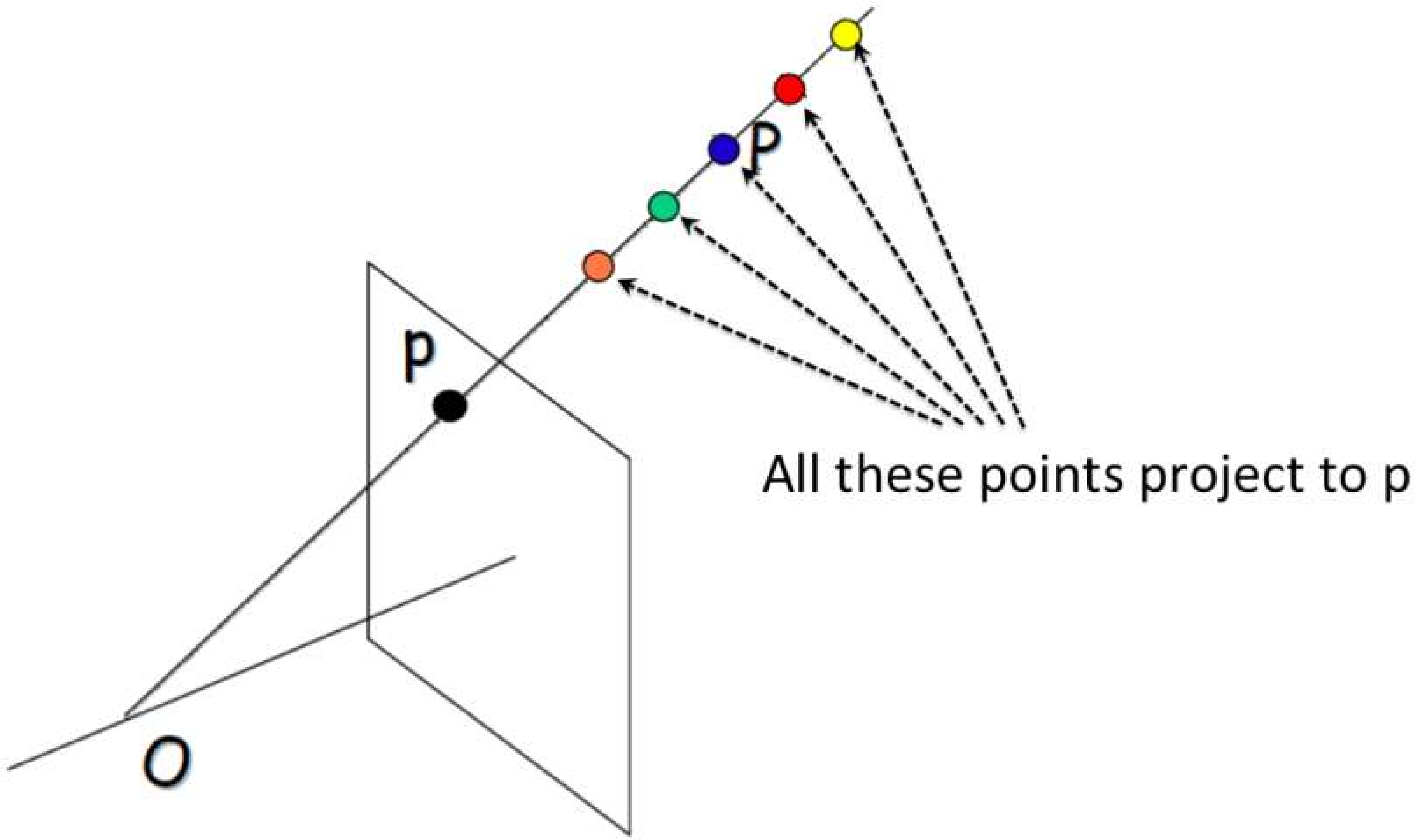
- Calibración de cámara



Profundidad desde estéreo

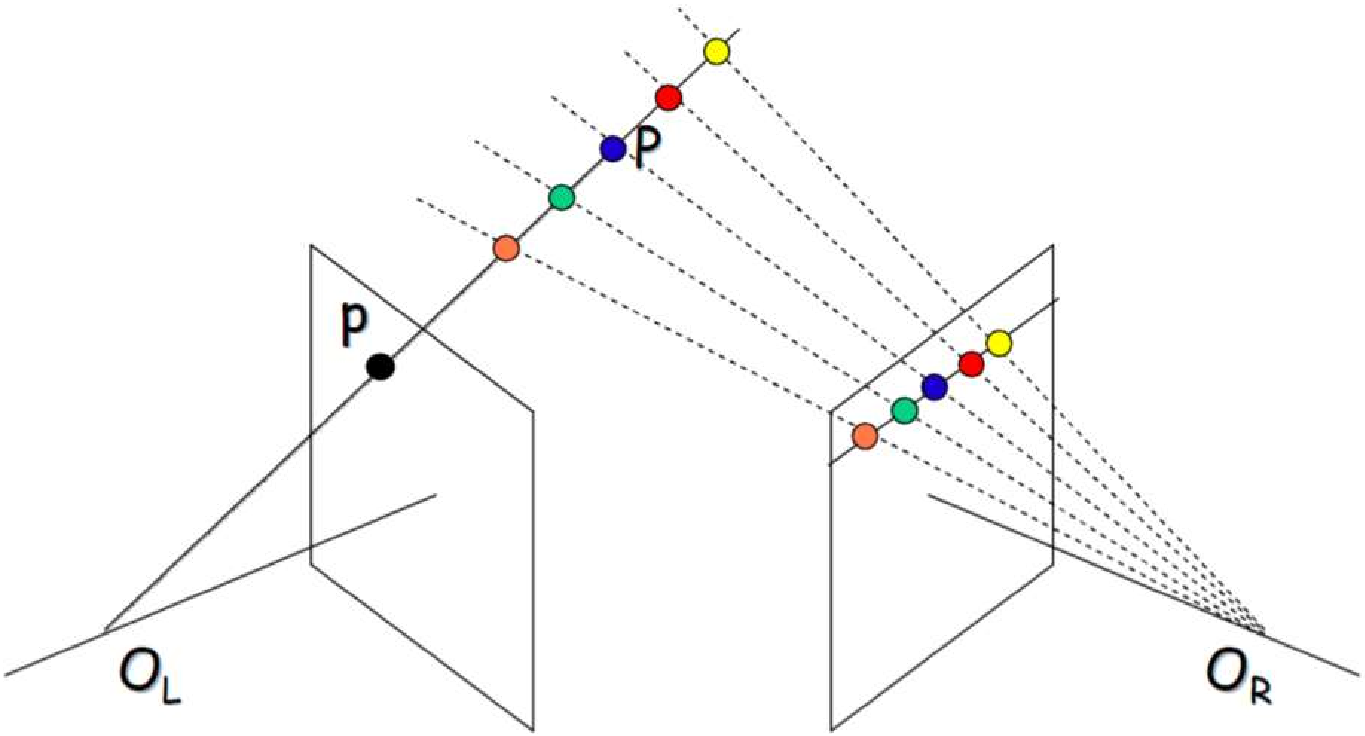
Profundidad desde dos vistas

- Todos los puntos en la línea proyectiva a P mapean a p



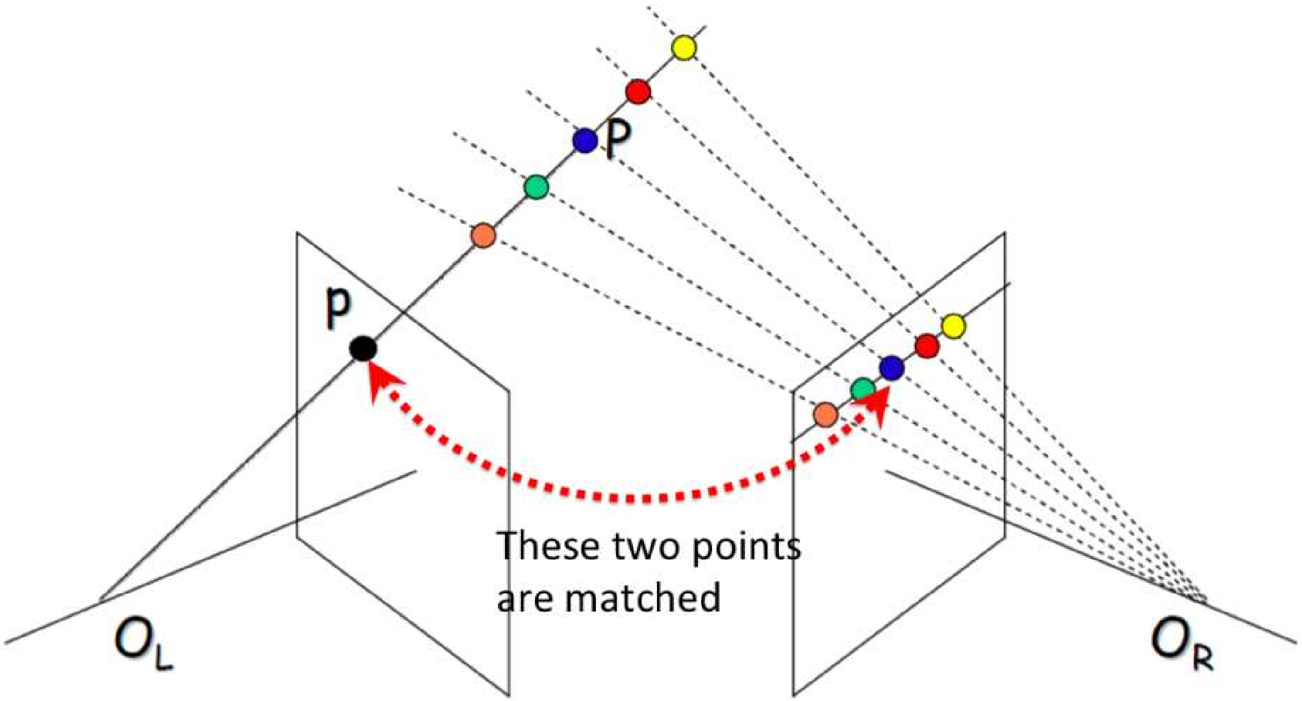
Profundidad desde dos vistas

- Todos los puntos en la línea proyectiva a P en la cámara de la izquierda mapean a una línea en la imagen de la derecha



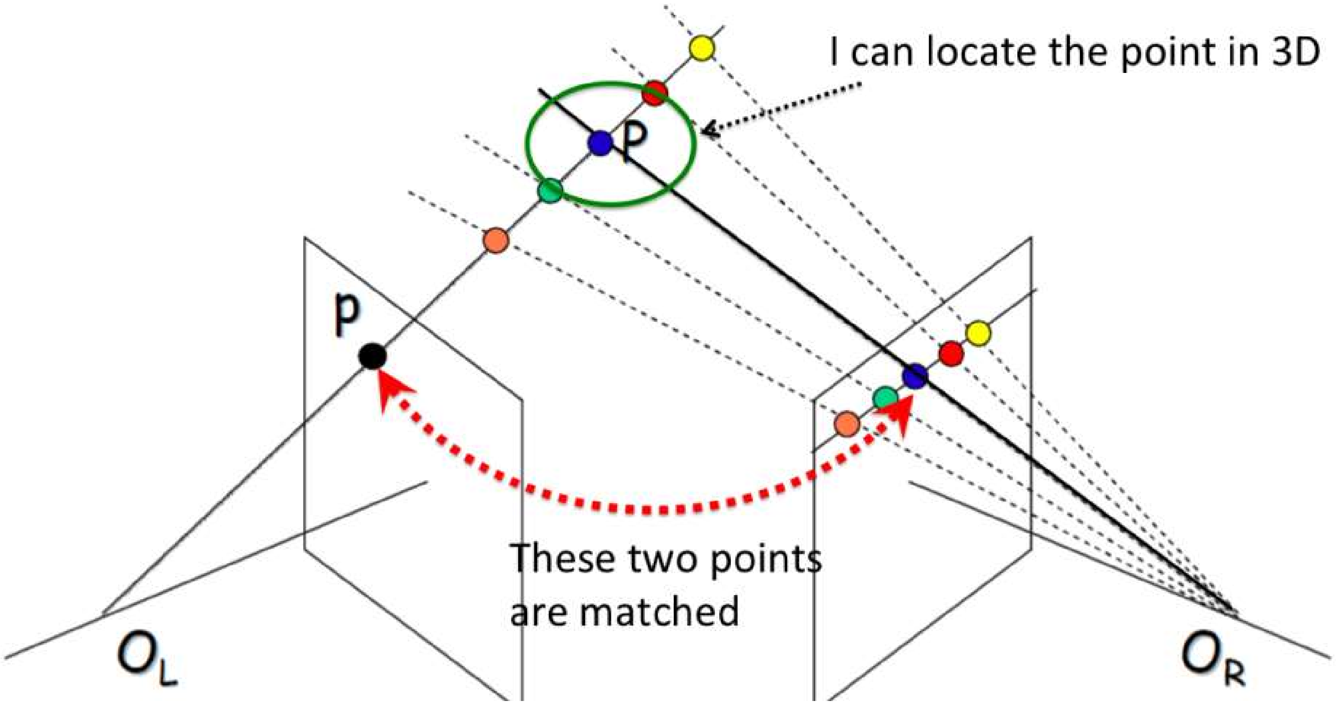
Profundidad desde dos vistas

- Si yo busco esta línea para encontrar correspondencias…



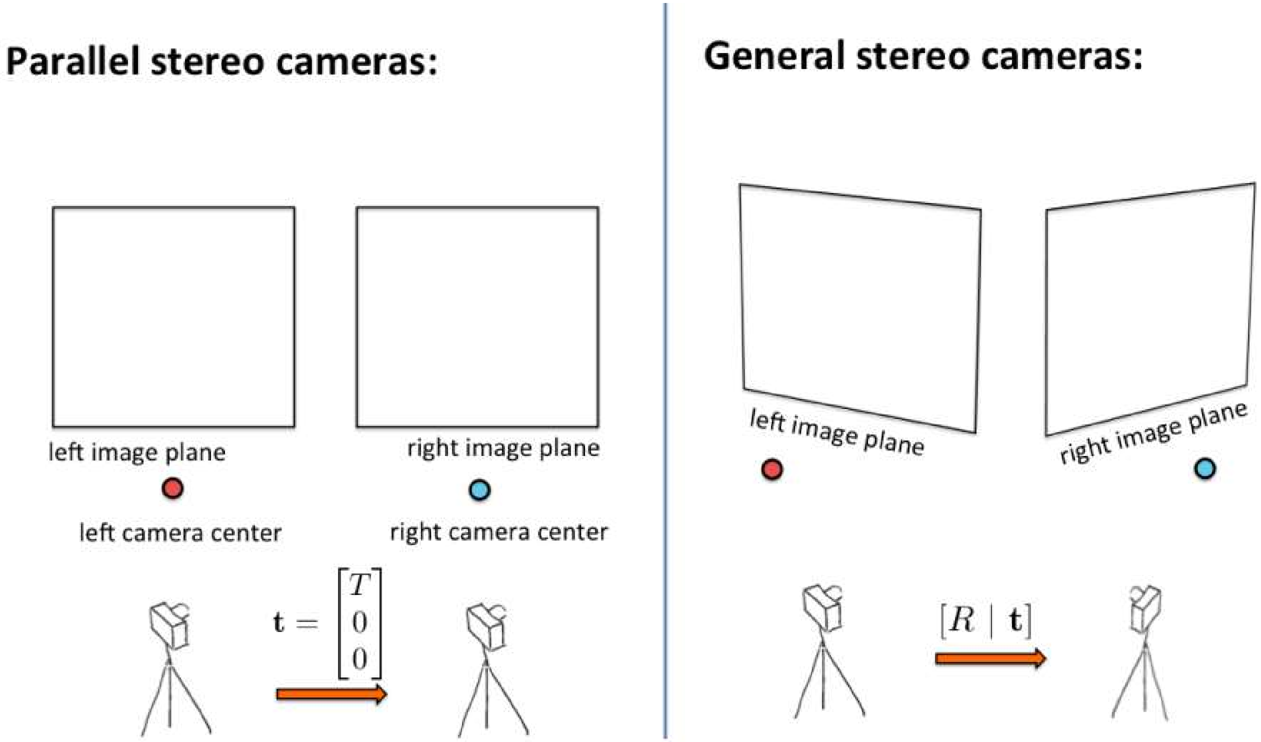
Profundidad desde dos vistas

- Podemos ubicar puntos 3D!



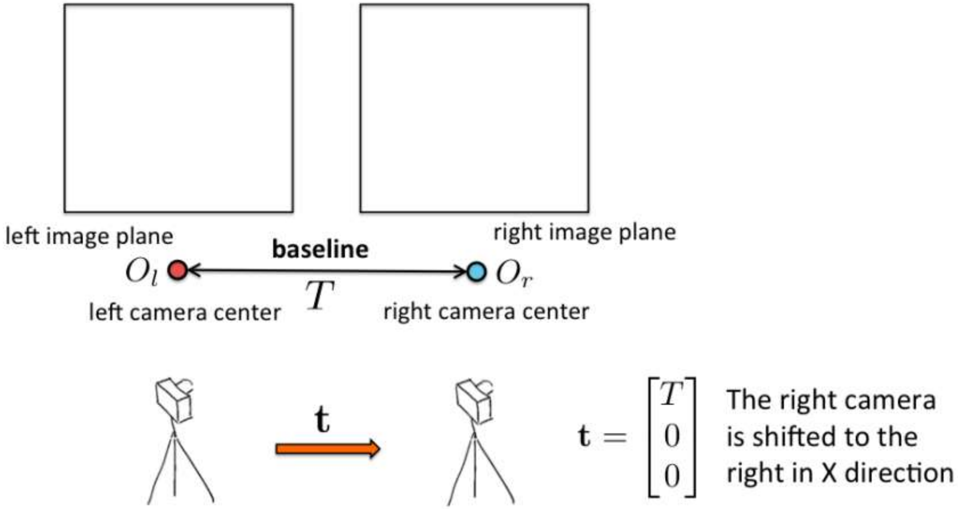
Geometría epipolar

- Caso con dos cámaras con ejes ópticos paralelos
- Caso general



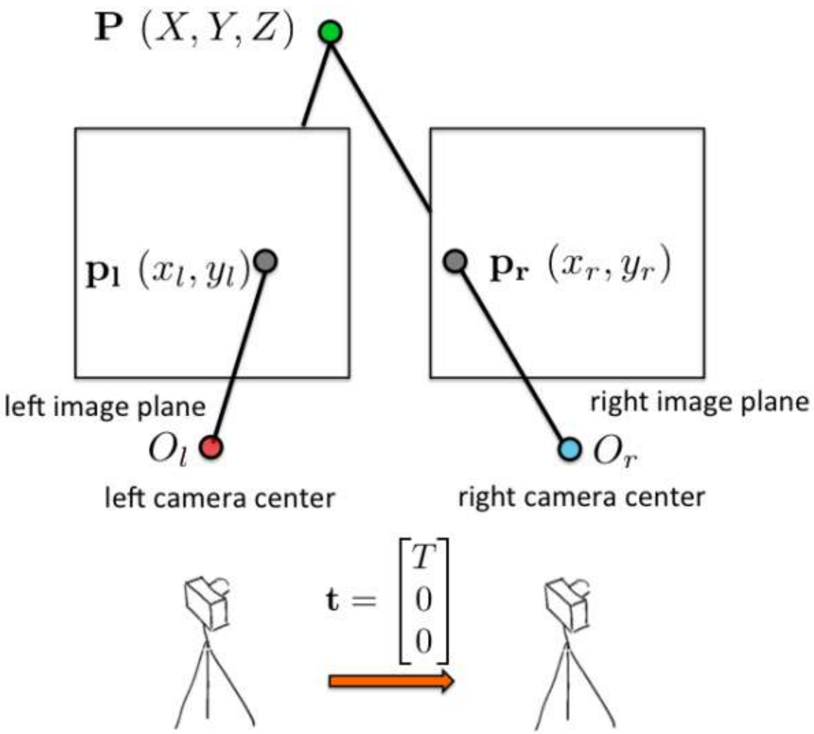
Cámaras paralelas

- Asumimos que las dos cámaras calibradas (se conocen extrínsecos y intrínsecos) son paralelas. Cámara de la derecha está a alguna distancia conocida de la cámara de la izquierda. Lo llamamos baseline.



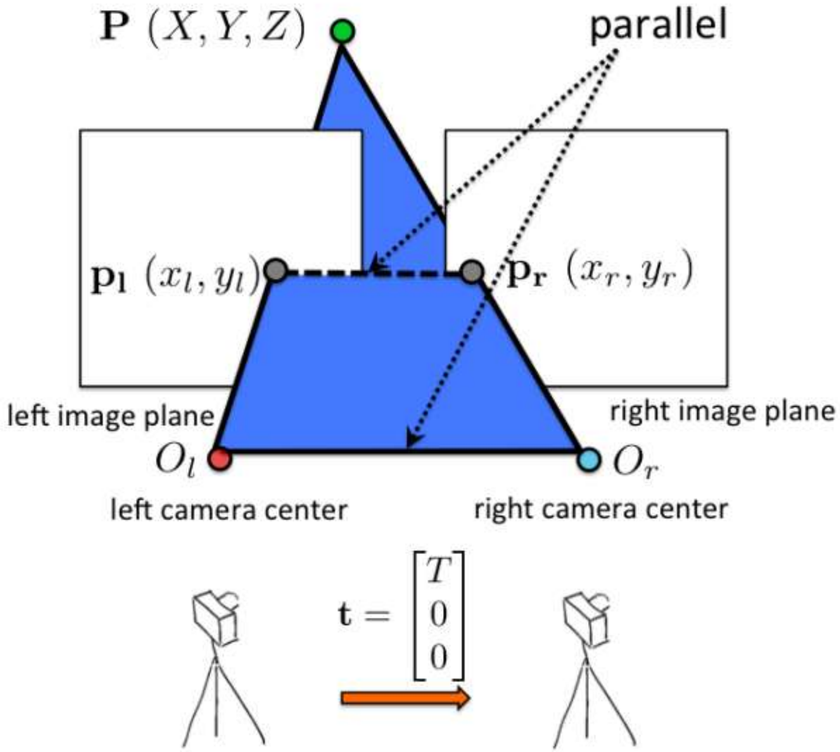
Cámaras paralelas

- Escoger un punto P en el mundo



Cámaras paralelas

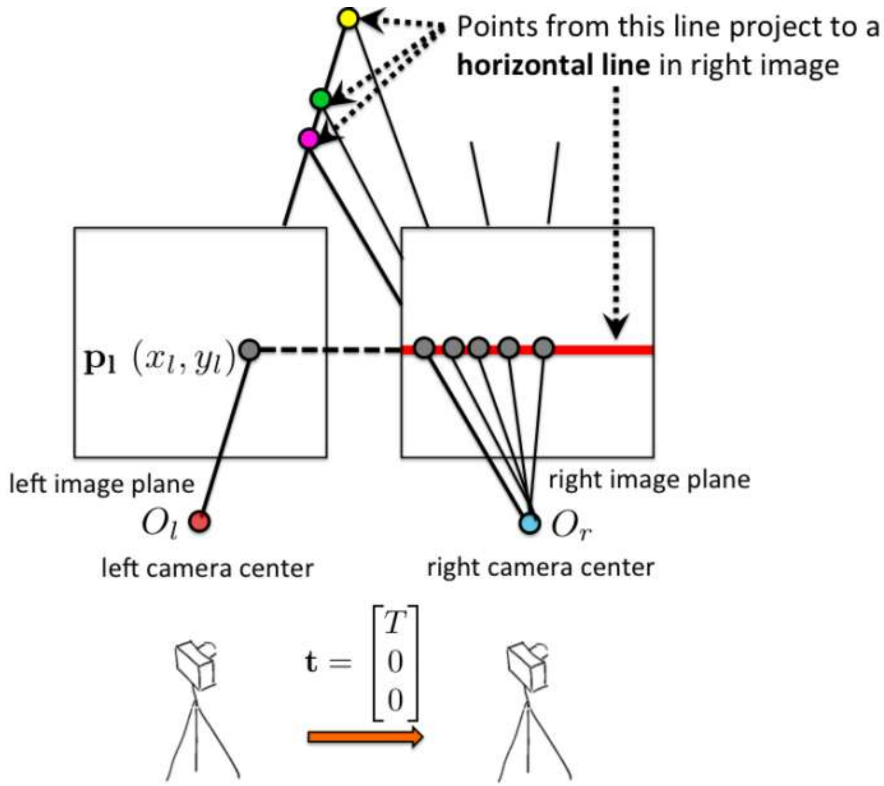
- Puntos O_l, O_r y P $(p_l\ y\ p_r)$ caen en un plano. Los planos de imagen son el mismo, entonces las líneas $O_l O_r$ y $p_l p_r$ son paralelos.



$$\boxed{\text{So:} \quad y_r = y_l}$$

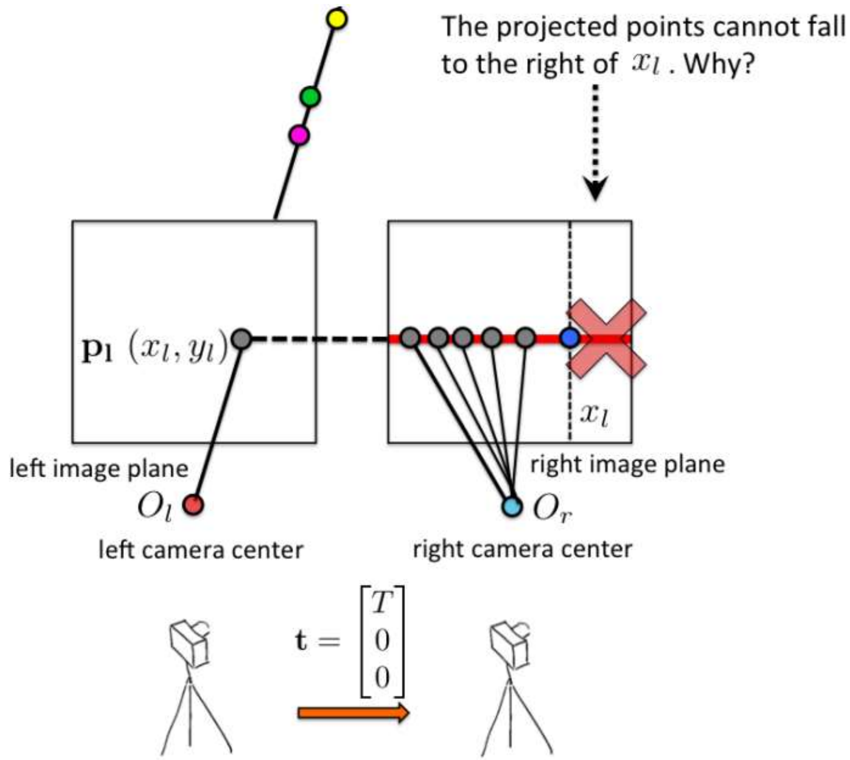
Cámaras paralelas

- Todos los puntos en la línea proyectiva $O_l p_l$ se proyecta a una línea horizontal con $y = y_l$ en la imagen de la derecha.



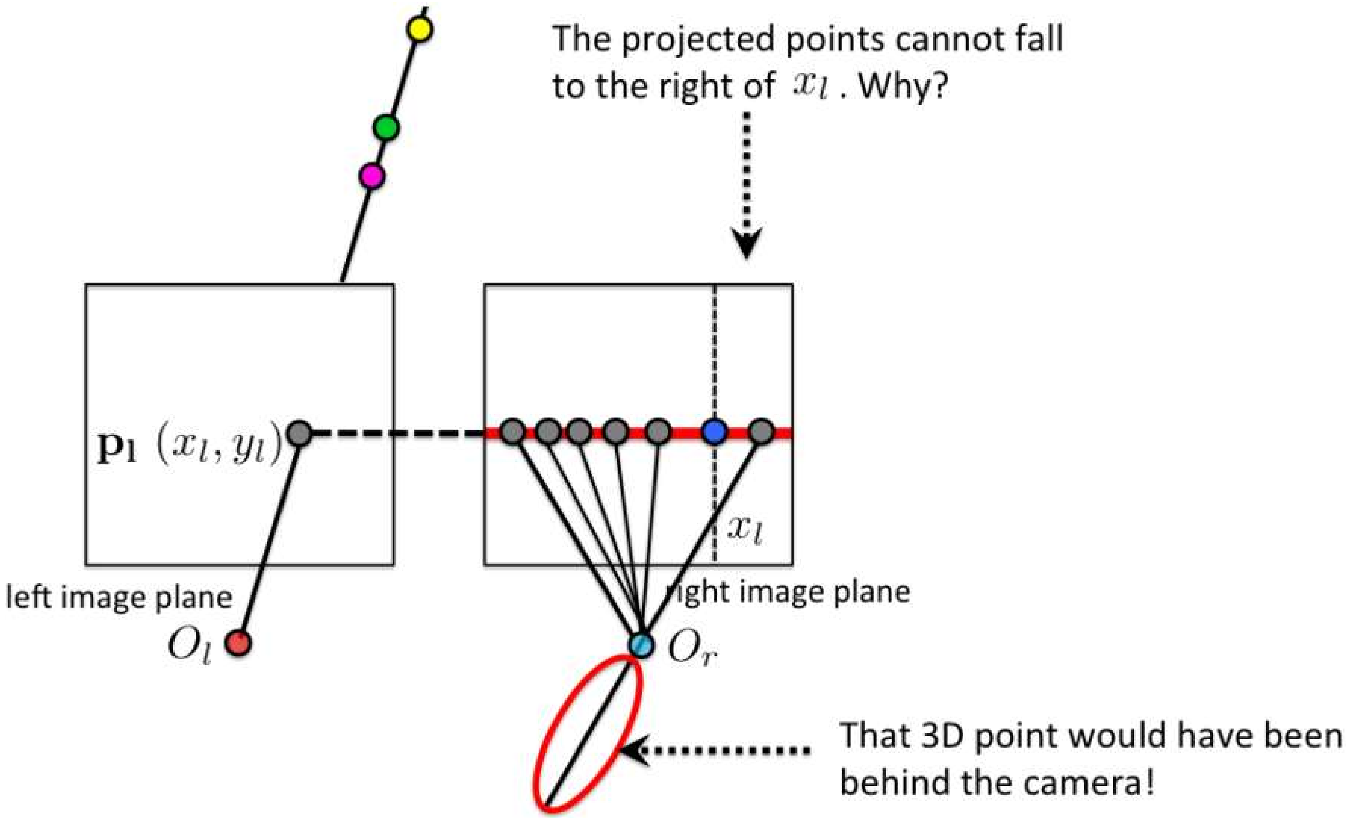
Cámaras paralelas

- Otra observación: Ningún punto en $O_l p_l$ se puede proyectar a la derecha de x_l en la imagen de la izquierda



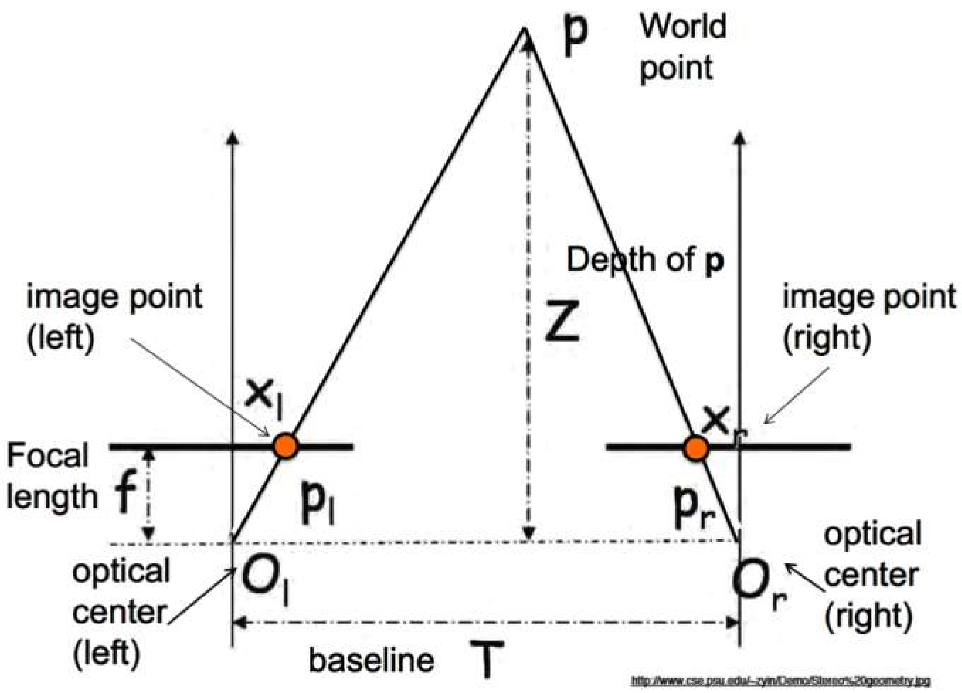
Cámaras paralelas

- Porque significa que la imagen puede ver atrás de la cámara



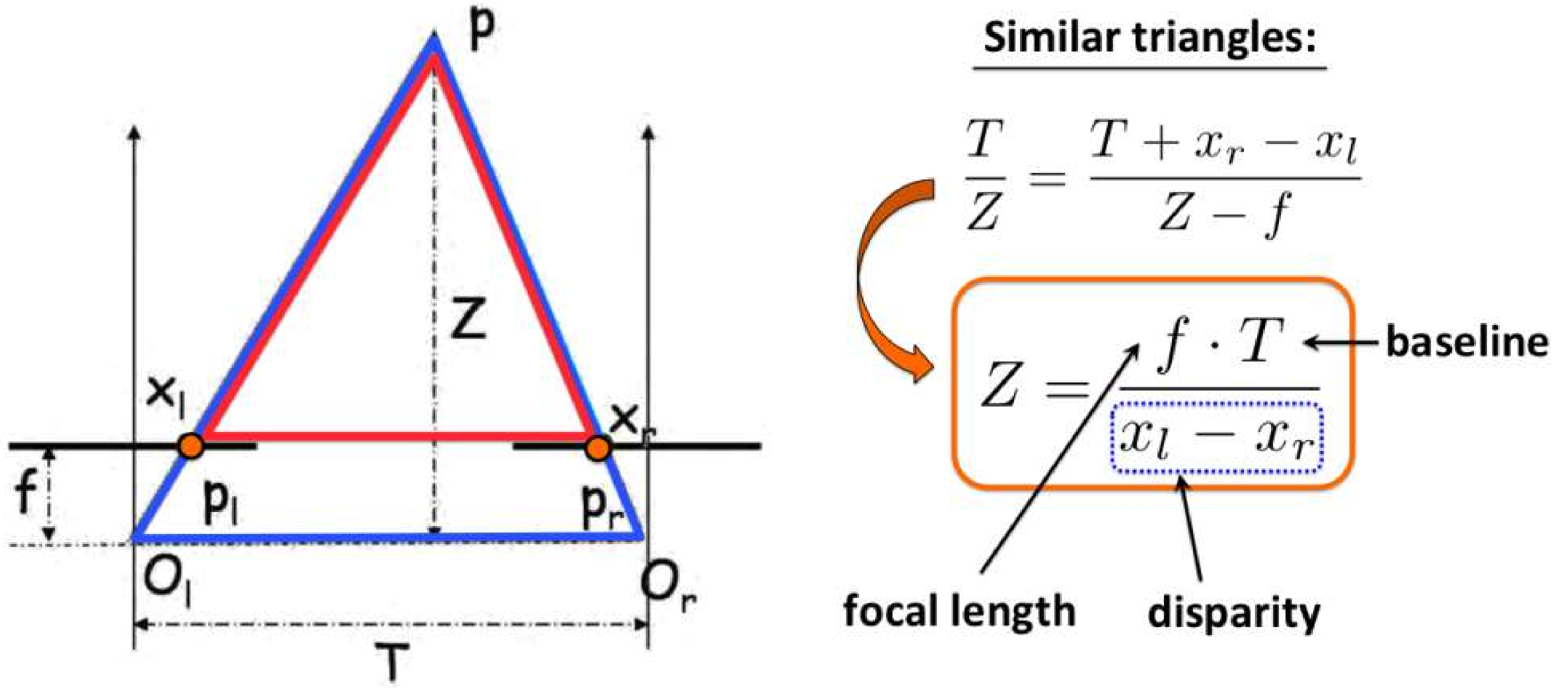
Cámaras paralelas

- Como los puntos p_l y p_r caen en una línea horizontal. Miremos la cámara desde arriba



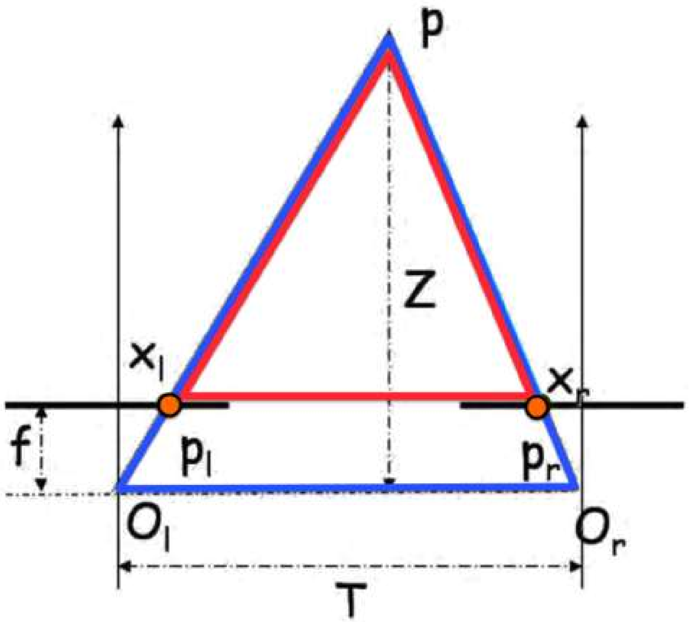
Cámaras paralelas

- Usamos triángulos similares para computar la profundidad del punto P



Cámaras paralelas

- Usamos triángulos similares para computar la profundidad del punto P



Similar triangles:

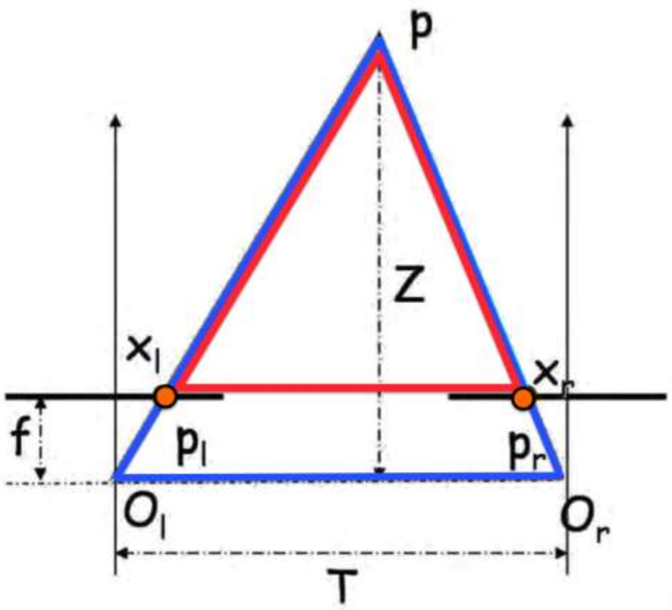
$$\frac{T}{Z} = \frac{T + x_l - x_r}{Z - f}$$

$$\boxed{Z = \frac{f \cdot T}{x_r - x_l}}$$

So if I know x_l and x_r, then I can compute Z!

Cámaras paralelas

- Usamos triángulos similares para computar la profundidad del punto P



Similar triangles:

$$\frac{T}{Z} = \frac{T + x_l - x_r}{Z - f}$$

$$\boxed{Z = \frac{f \cdot T}{x_r - x_l}}$$

$$x = \frac{f \cdot X}{Z} + p_x$$

And if I know Z, I can compute X and Y, which gives me the point in 3D

Cámaras paralelas

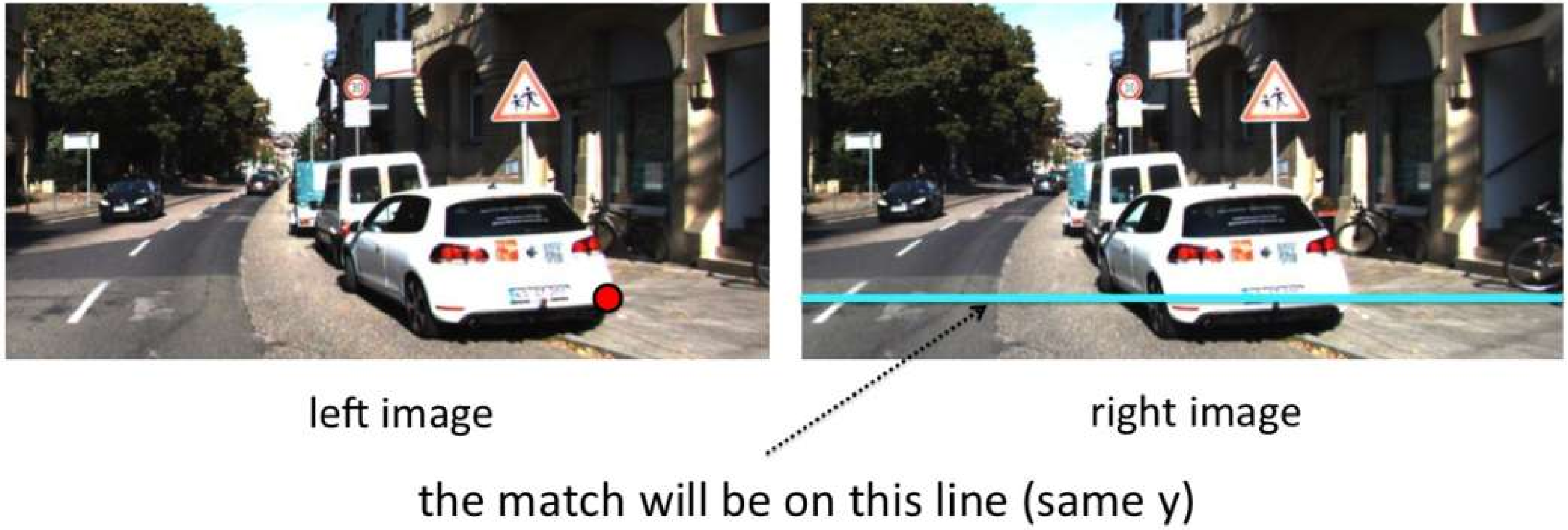
- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?

left image

right image

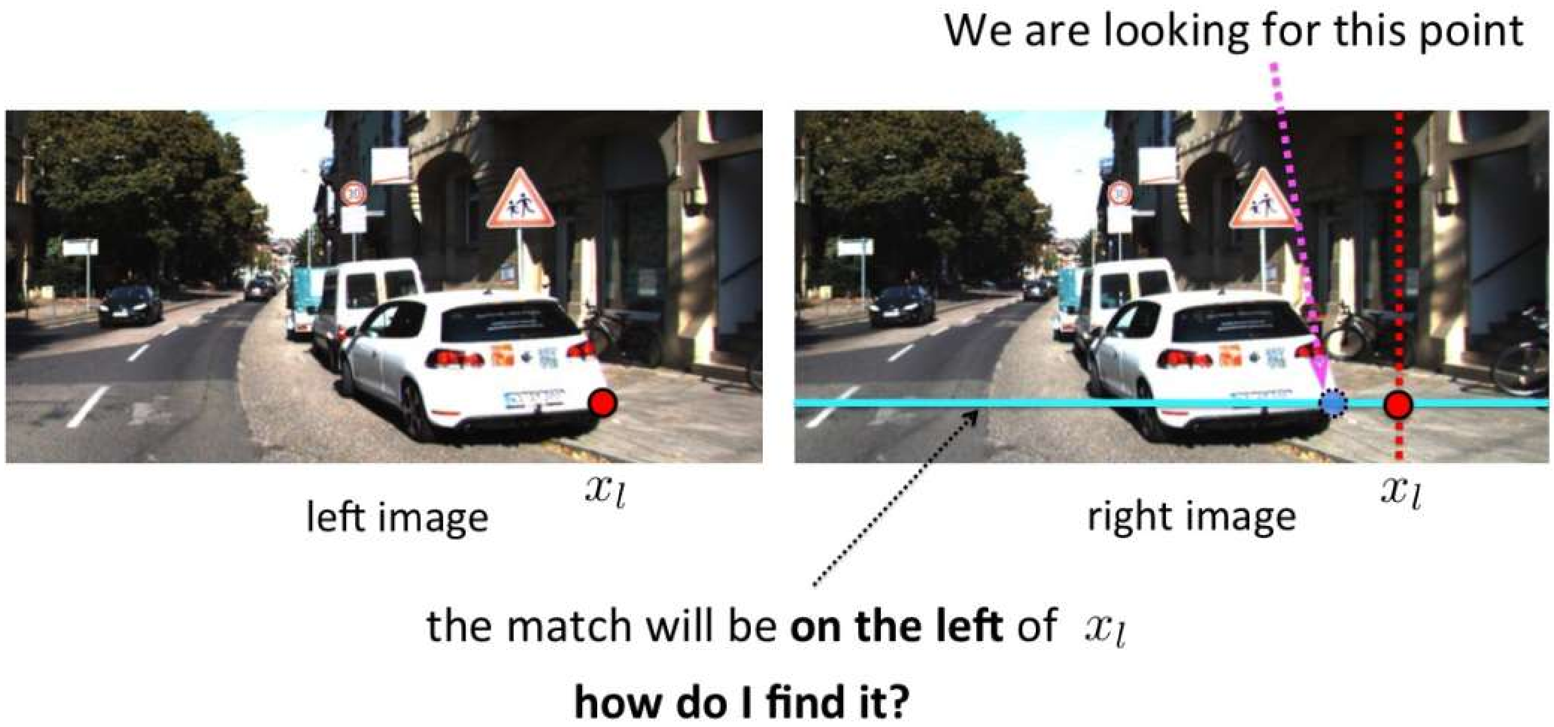
Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$



Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$



Cámaras paralelas

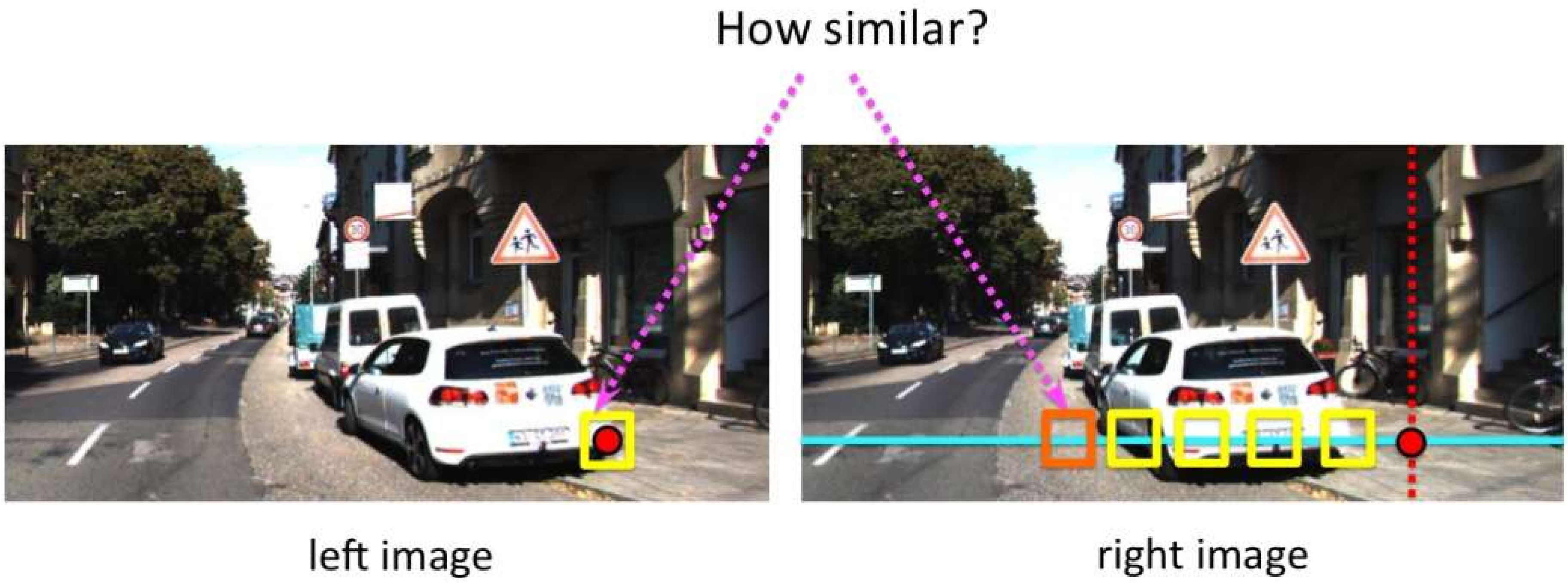
- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$

left image

right image

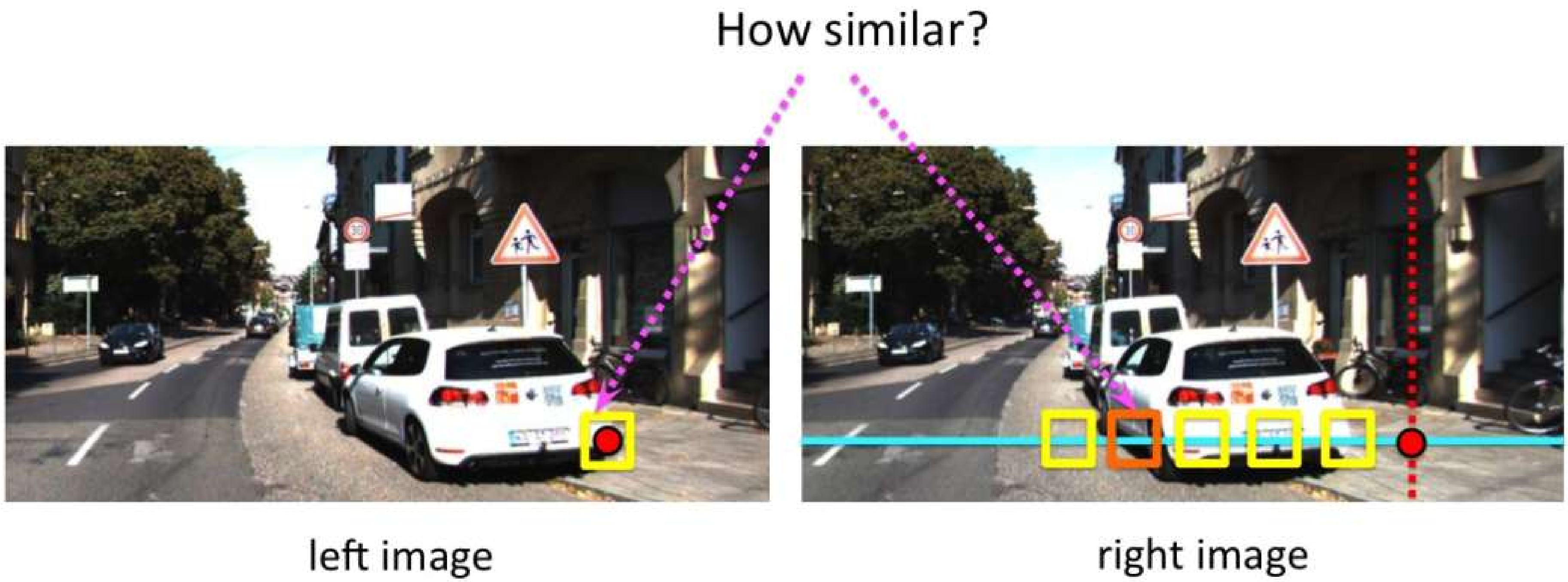
Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$



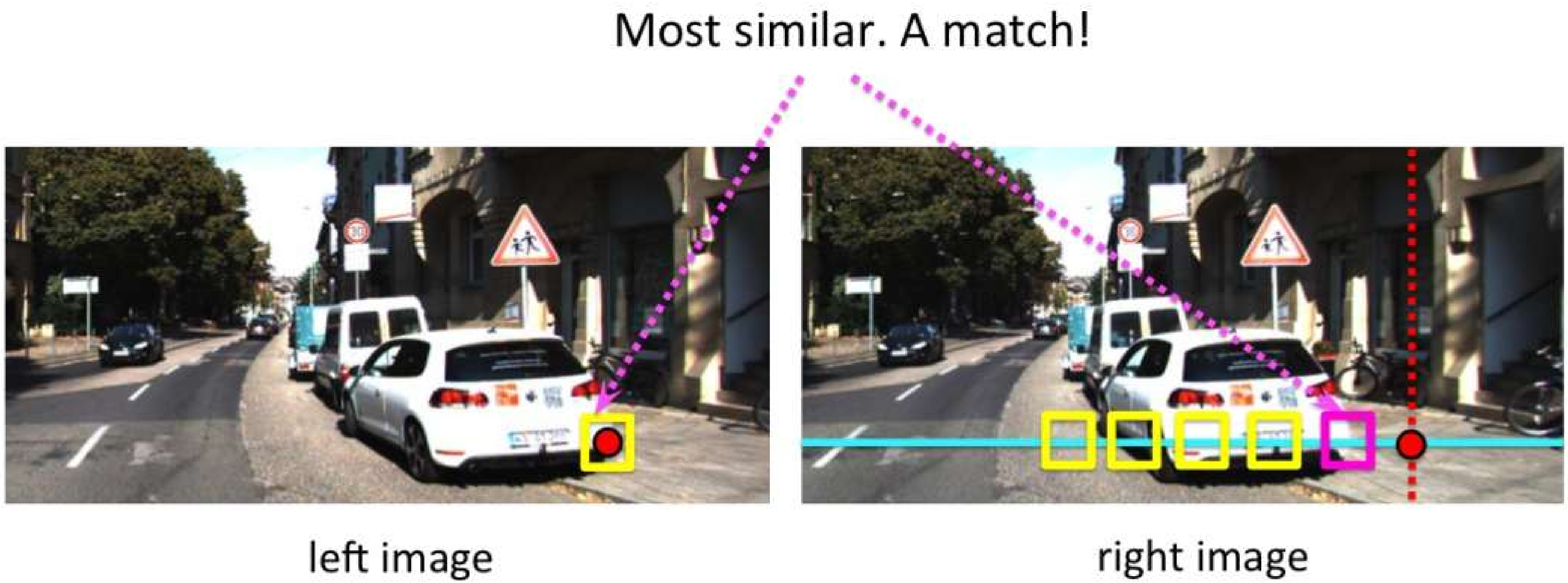
Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$



Cámaras paralelas

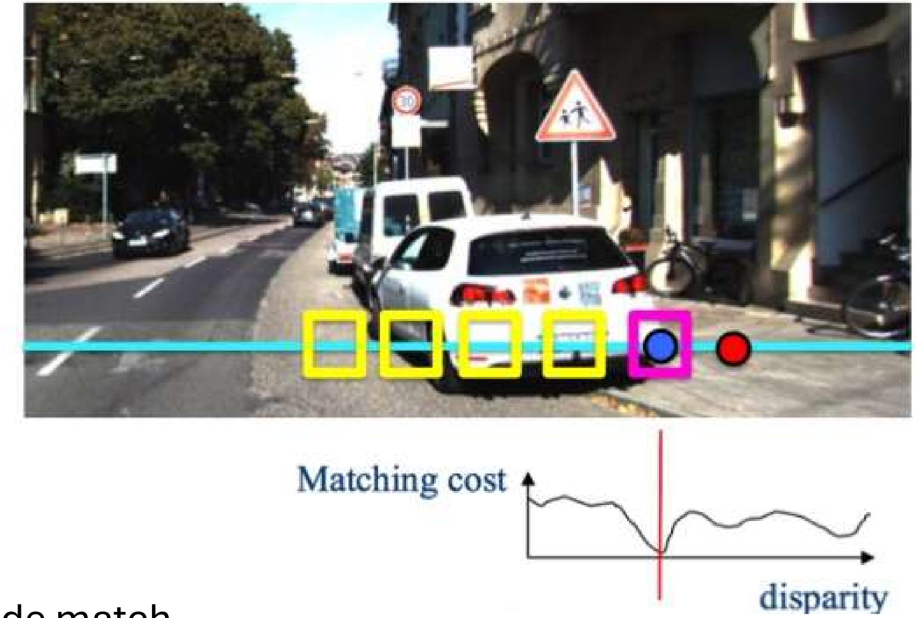
- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$



Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$

left image



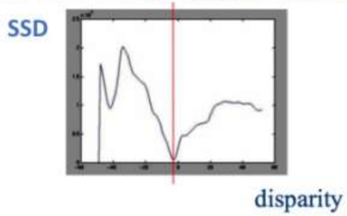
En cada punto del scanline, se computa un costo de match (correlación cruzada, por ejemplo)

Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$

$$SSD(\text{patch}_l, \text{patch}_r) = \sum_x \sum_y (I_{\text{patch}_l}(x, y) - I_{\text{patch}_r}(x, y))^2$$

left image



Compute a matching cost

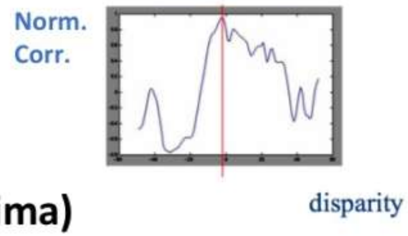
Matching cost: **SSD** **(look for minima)**

Cámaras paralelas

- Dado un punto $p_l = (x_l, y_l)$, cómo obtenemos $p_r = (x_r, y_r)$?
- Matcheamos en línea $y_r = y_l$

$$NC(\text{patch}_l, \text{patch}_r) = \frac{\sum_x \sum_y (I_{\text{patch}_l}(x,y) \cdot I_{\text{patch}_r}(x,y))}{||I_{\text{patch}_l}|| \cdot ||I_{\text{patch}_r}||}$$

left image

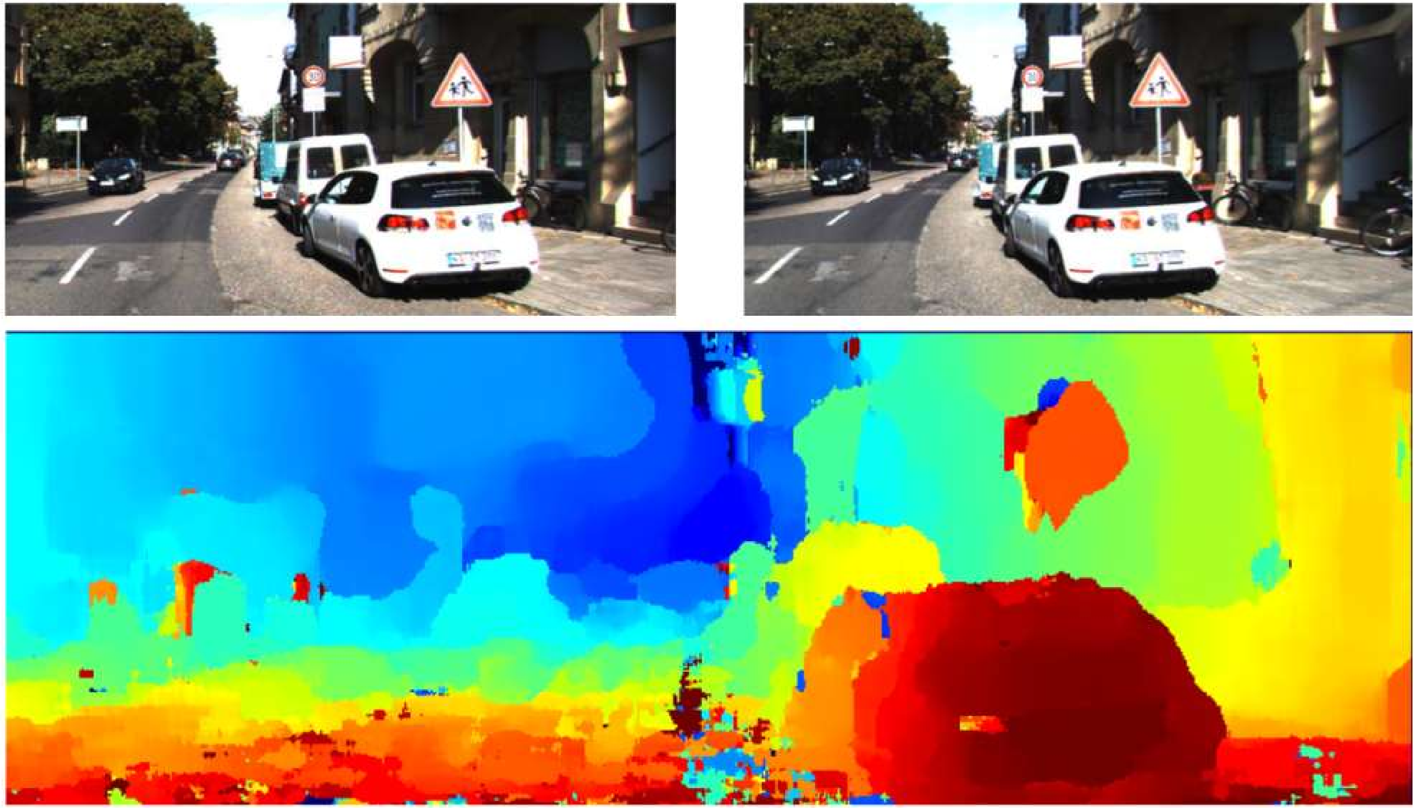


Compute a matching cost

Matching cost: **Normalized Corr. (look for maxima)**

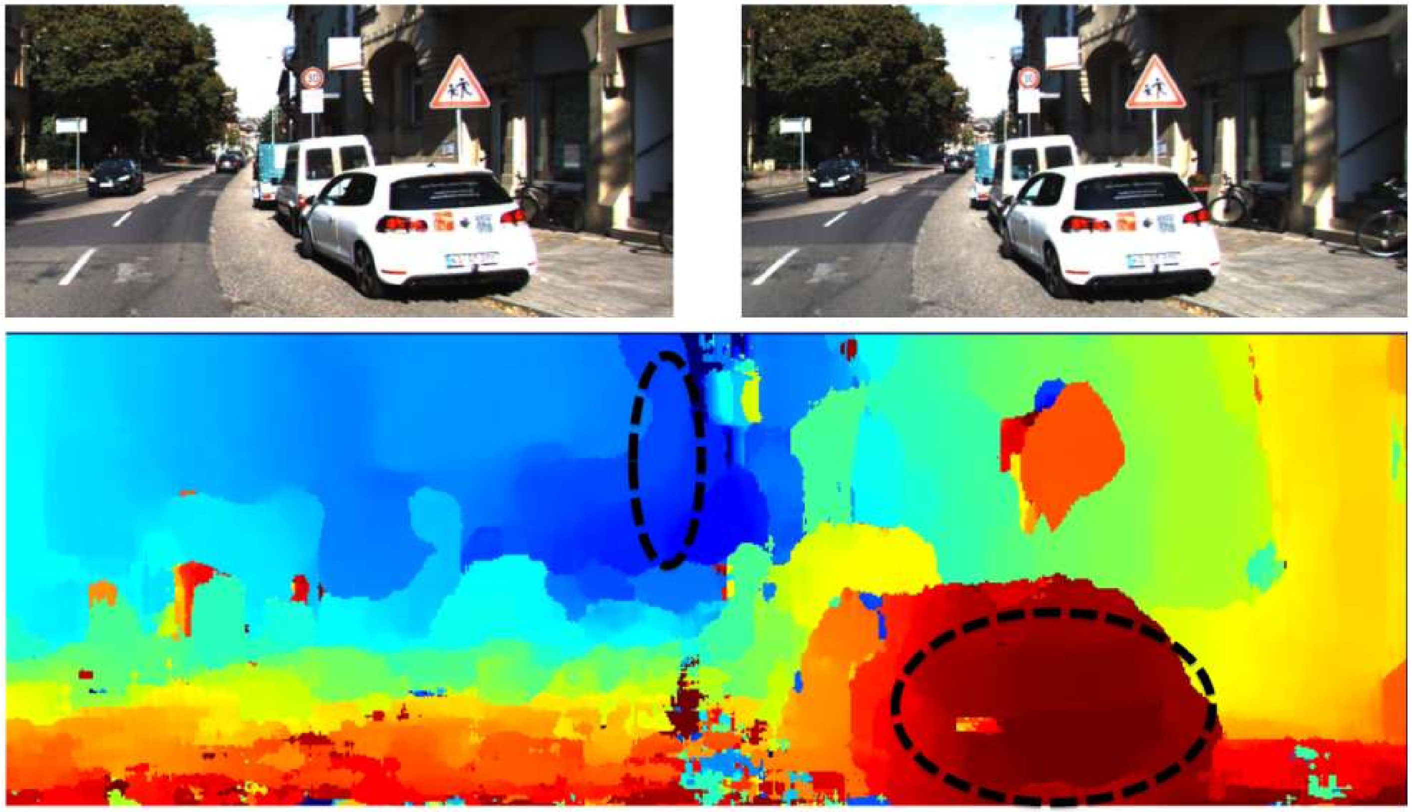
Cámaras paralelas

- Se puede obtener un mapa de disparidad



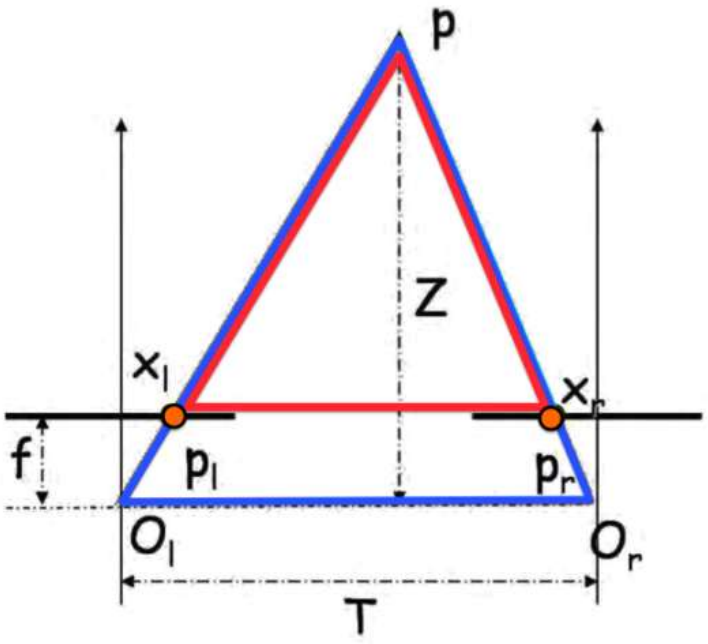
Cámaras paralelas

- Objetos más cercanos tienen alta disparidad. Objetos lejanos tienen baja disparidad



Cámaras paralelas

- Objetos más cercanos tienen alta disparidad. Objetos lejanos tienen baja disparidad



Similar triangles:

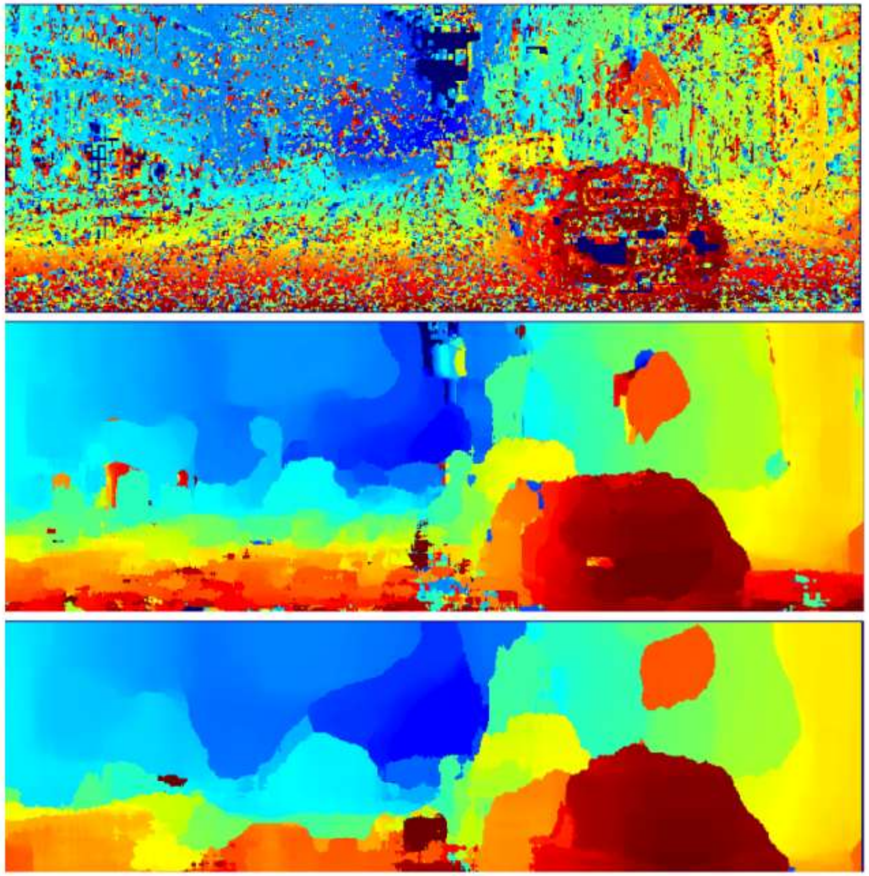
$$\frac{T}{Z} = \frac{T + x_l - x_r}{Z - f}$$

$$\boxed{Z = \frac{f \cdot T}{\boxed{x_r - x_l}}}$$

Depth (Z) and disparity are **inversely proportional**

Cámaras paralelas

- Tamaño del bloque de comparación



patch size $= 5$

patch size $= 35$

patch size $= 85$