

Máster Universitario en
Nuevas Tecnologías en Informática

Asignatura “Visión Artificial”

Geometría Proyectiva I

Formación de imágenes

Modelo lineal de cámara

Calibración extrínseca/intrínseca

Rectificación de planos

Facultad de Informática
Universidad de Murcia
Curso 2018/19

Introducción a la geometría proyectiva

- **Objetivos principales:**

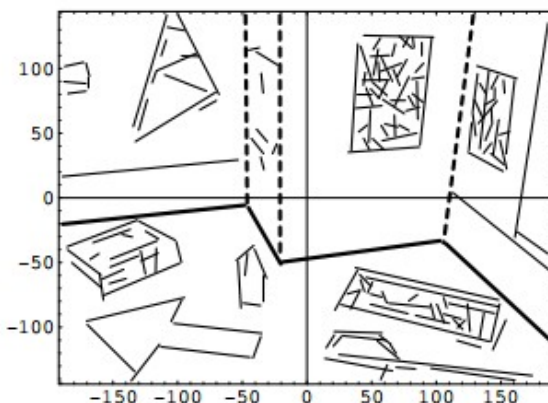
- Extracción de “estructura espacial del mundo” a partir de una (***monocular reconstruction***) o múltiples imágenes (***SfM, Structure From Motion***).
- También, subsidiariamente, la posición de la cámara respecto a la escena.
- Operar en ambientes progresivamente más difíciles (sin *landmarks*, parcialmente no estructurados, etc.)
- En algún sentido, **problema inverso a los gráficos 3D por computador**.
- “Biblia” de la geometría proyectiva: [Hartley & Zisserman 2003].

Reconstrucción 3D

- Ejemplo de reconstrucción monocular:



Imagen original



Extracción y clasificación de *features* (segmentos, con segmentos de separación de planos resaltados)



Reconstrucción tridimensional

Reconstrucción estéreo

- Ejemplo de reconstrucción a partir de dos vistas:



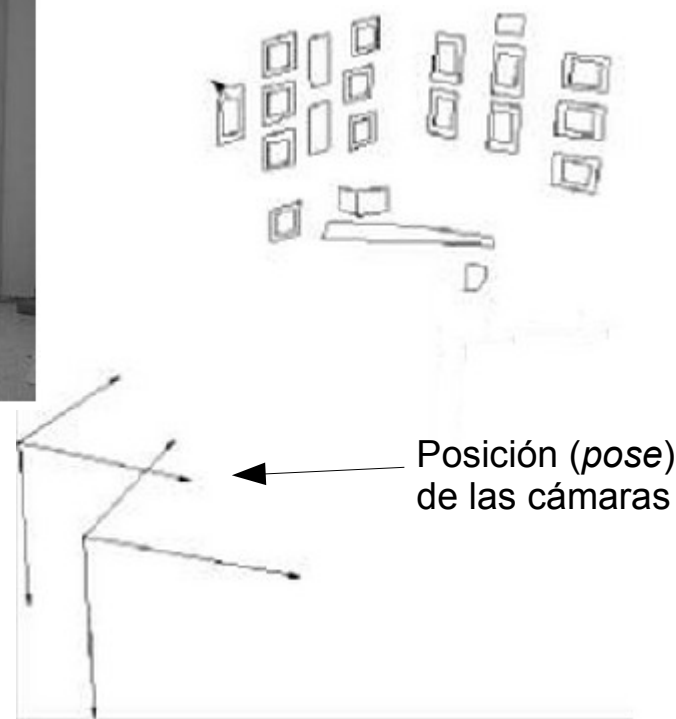
Imagen izquierda



Imagen derecha



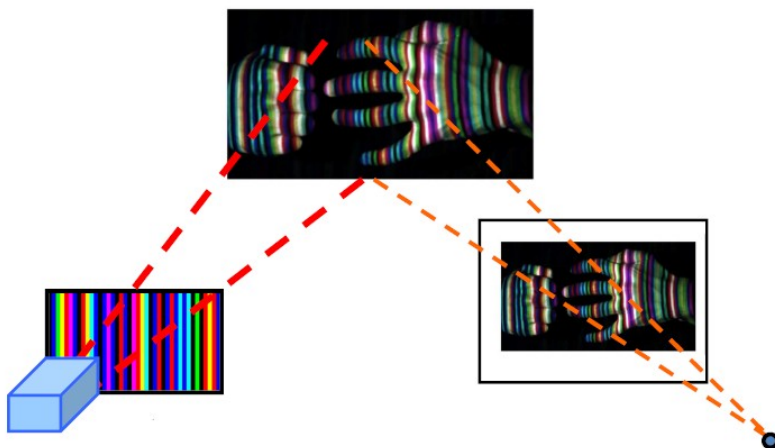
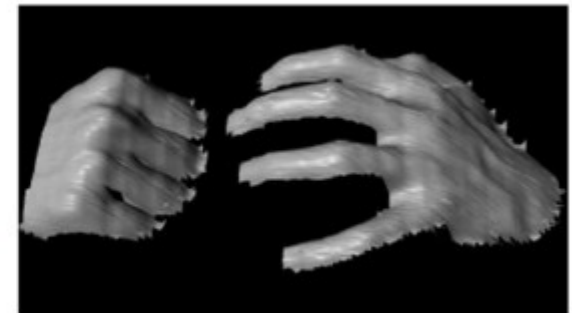
Ejemplo de reconstrucción densa (mapa de profundidad)



Reconstrucción “sparse”
(sólo *features*)

Estéreo activo

- Ejemplos de reconstrucción con luz estructurada:
 - Idea subyacente en la Kinect de XBOX, sólo que con luz infraroja (no visible).

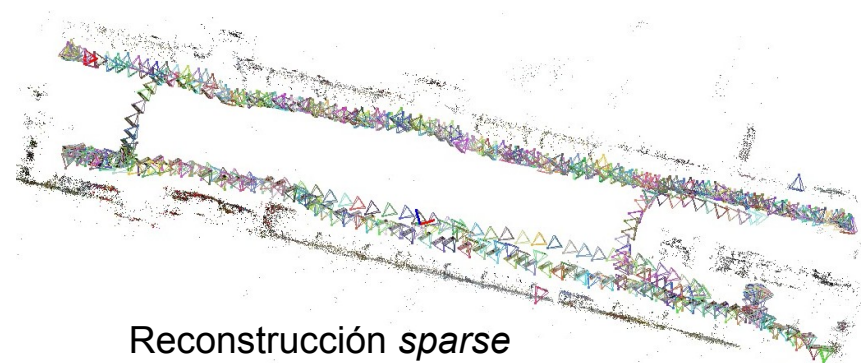


Secuencias de vídeo

- Ejemplo de reconstrucción a partir de secuencias de vídeo:



Secuencia de vídeo



Reconstrucción *sparse*



Reconstrucción *densa*

El *matching* de features entre imágenes es más sencillo, al conocerse la secuencia temporal de las mismas (se puede hacer *tracking* continuo)

Múltiples vistas

- Ejemplo de reconstrucción a partir de múltiples vistas:



Colección de fotos
“desestructurada”



Reconstrucción
tridimensional y poses
de las cámaras

Realidad aumentada

- Ejemplo de realidad aumentada utilizando patrones (*landmarks*):
 - Implica cálculo de *pose* cámara – escena



Formación de las imágenes

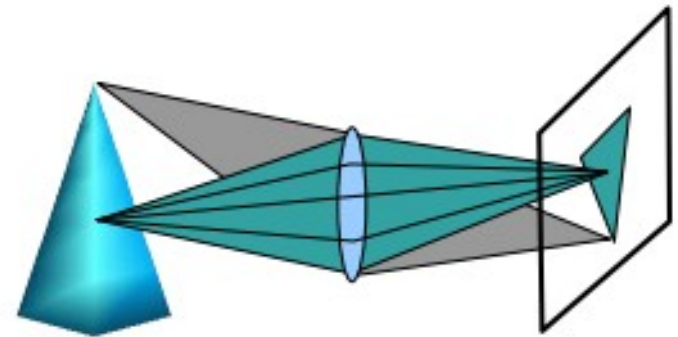
- Modelo de cámara tipo *pinhole*:
 - Cada punto 3D de la escena genera un rayo a través del centro óptico de la cámara, que intersecta al plano de imagen en un punto 2D proyectado.



Rayos a través de
agujero muy pequeño



cf. John Barrow:
Perspectiva:
Grabado sobre madera perteneciente a
Mujeres de la Naturaleza de J.M. London, 1635.

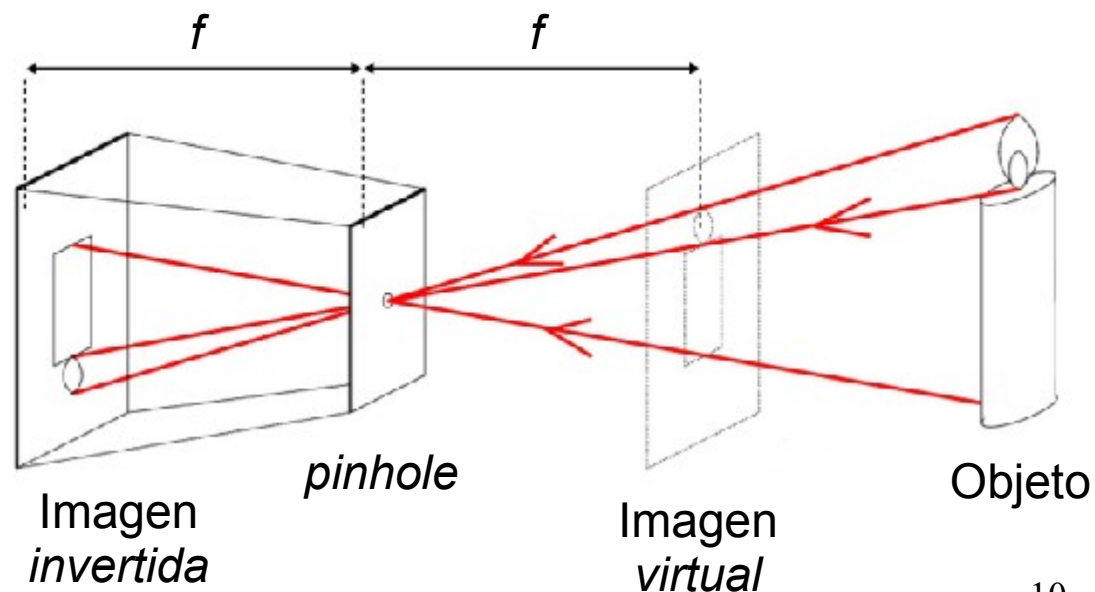
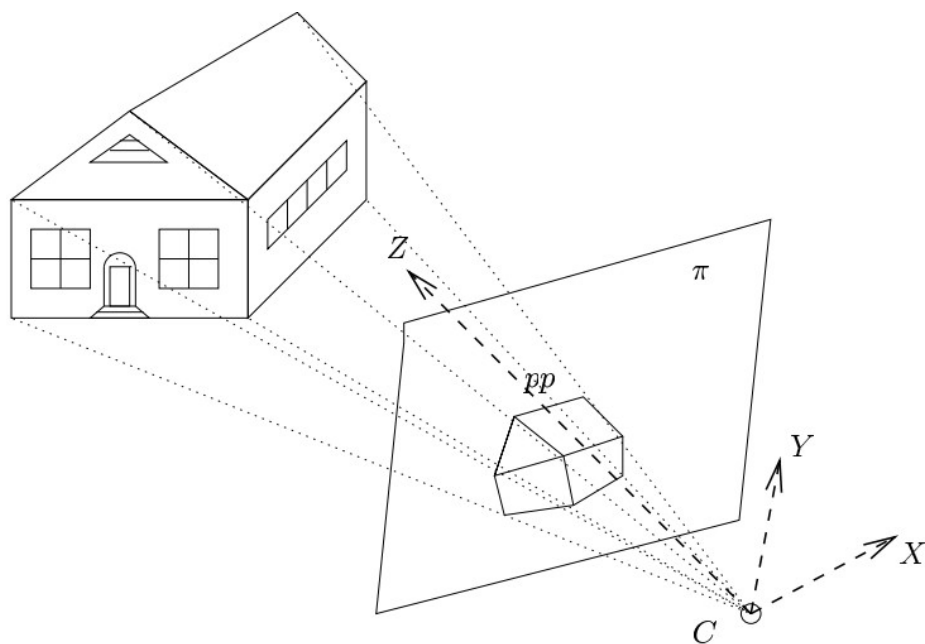


Lente (idéntico efecto,
pero consigue más luz)

Surgen temas de enfoque,
profundidad de campo, etc.

Transformación perspectiva

- Una cámara queda determinada por las posiciones de su centro óptico C y su plano de imagen π .
- El plano π está a una distancia (focal) f de C .
- La cámara mira en la dirección perpendicular desde C hasta π , y dicha dirección corta a π en el llamado *punto principal*.



(Mismo principio que la cámara oscura)

Ecuaciones de perspectiva (I)

- Si C estuviese en el origen $(0,0,0)$, y la cámara apuntase simplemente en la dirección del eje Z :
 - Como un escalado de coordenadas, pero con factor inversamente proporcional a la distancia perpendicular al plano de imagen:

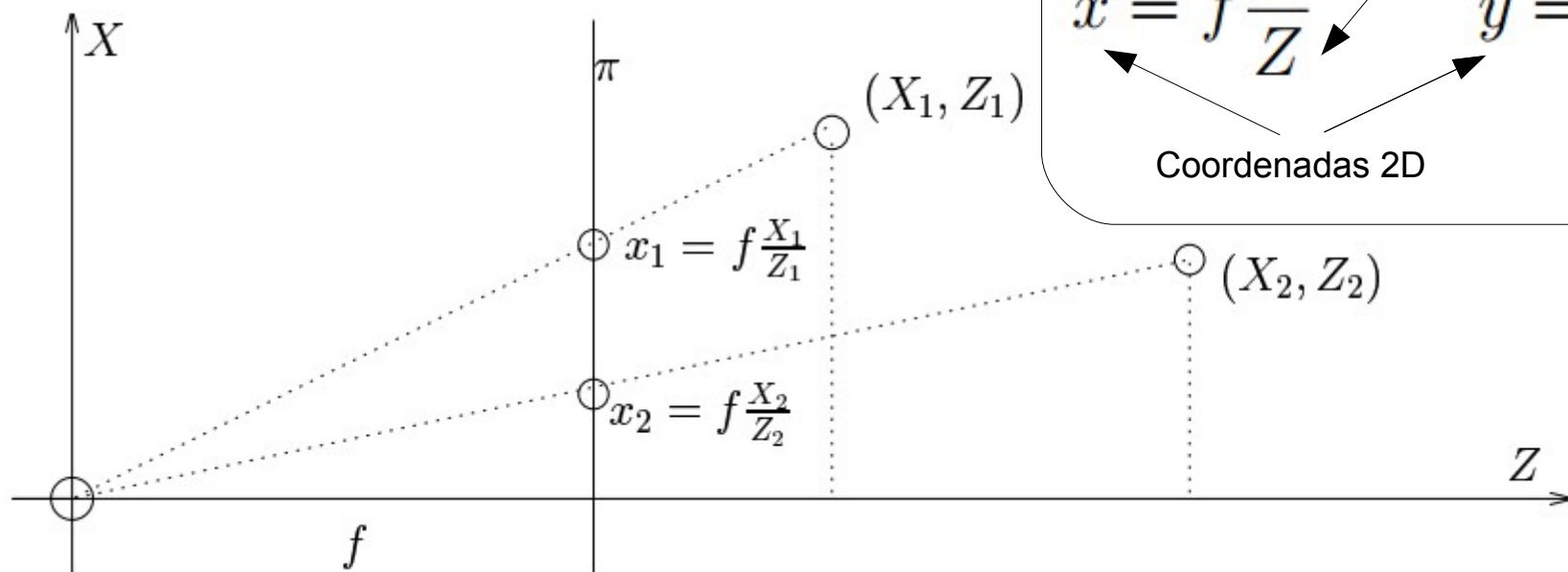


Diagram illustrating the perspective projection equations:

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

Labels in the diagram:

- Coordenadas 3D (pointing to X and Y)
- Coordenadas 2D (pointing to x and y)
- Distancia focal (pointing to f)

Ecuaciones de perspectiva (II)

- A pesar de la no linealidad de la división por Z , la mayor parte de las técnicas que usaremos estarán basadas simplemente en el álgebra lineal.
- Truco: coordenadas homogéneas:

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

*Coordenadas de imagen
homogéneas*

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

*Coordenadas de escena
homogéneas*

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

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

*Conversión a/desde coordenadas
homogéneas*

Ecuaciones de perspectiva (III)

- El truco de expresar los puntos 2D y 3D con coordenadas homogéneas permite expresar las traslaciones y transformaciones de perspectiva de forma **lineal** (las rotaciones y escalados ya lo son):

Simple multiplicación
matriz-vector

$$T = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} & & & 0 \\ R_{3 \times 3} & & & \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Traslación

Rotación

Escalado

Proyección 3D → 2D

Ejemplo de
proyección
con focal f :

$$\begin{bmatrix} f & x \\ f & y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \xrightarrow{\text{Transformación homogénea}} \begin{bmatrix} f \frac{x}{z} \\ f \frac{y}{z} \\ z \end{bmatrix}$$

Modelo lineal de cámara (I)

- **Parámetros extrínsecos:**

- La posición $C=(C_x, C_y, C_z)$ del centro de la cámara.
- La rotación tridimensional R (matriz ortogonal 3x3, e.d., siempre cumple que $R \cdot R^T = I$) entre sistema de coordenadas de la cámara y el de la escena.
- Éste último es en cierto sentido arbitrario (ej. sistema mundo vs. sistema robot):



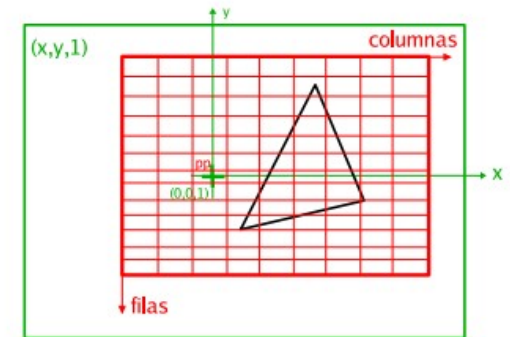
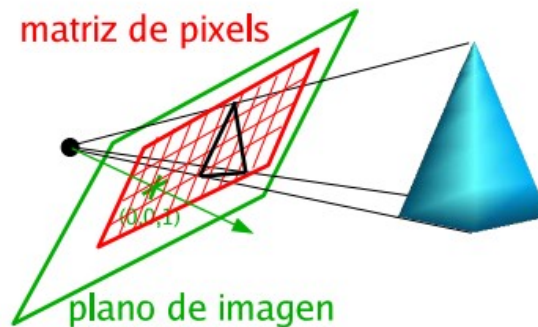
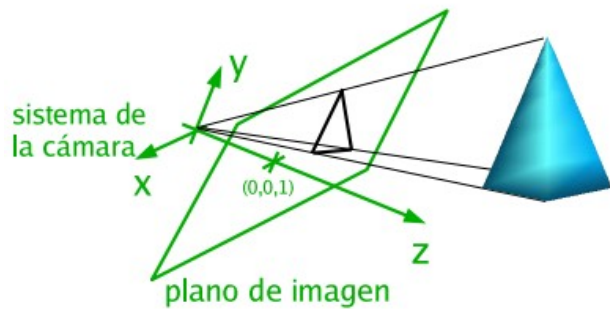
$C \rightarrow 3$ grados de libertad

$R \rightarrow 3$ grados de libertad
(aunque hay 9 elementos en R , sólo 3 ángulos *-pan, tilt, roll-* la determinan completamente)

$\Rightarrow 6$ gdl en total para posición de la cámara (parám. extrínsecos)

Modelo lineal de cámara (II)

- Parámetros intrínsecos:
 - Dependen únicamente de la cámara (no de su posición)
 - Modelan la transformación rayos del mundo \leftrightarrow píxeles de imagen



Coordenadas de píxel

$$\begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

=

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

Coordenadas de plano de imagen

$$K = \begin{bmatrix} f & s & o_x \\ 0 & fr & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Modelo lineal de cámara (III)

- Parámetros intrínsecos (cont.):

$$\begin{array}{ccc} \text{Coordenadas de píxel} & \longrightarrow & \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \longleftarrow \text{Coordenadas de plano de imagen} \\ & & \uparrow \\ & & \text{Matriz (triangular superior) de parámetros intrínsecos} \end{array}$$

Focal efectiva (distancia de C a π , en unidades de píxel)

Sesgo, o *skew* (admite posibilidad de píxeles no rectangulares)

Punto principal

Aspect ratio (admite posibilidad de píxel rectangular)

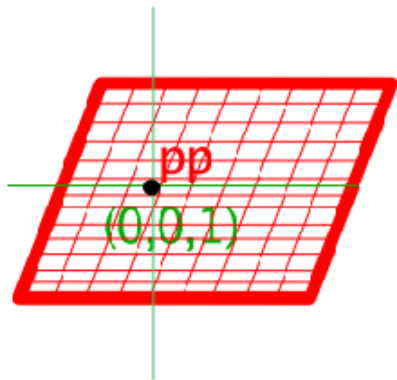
$$K = \begin{bmatrix} f & s & o_x \\ 0 & fr & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

\Rightarrow 5 *gdl* en total para parámetros Intrínsecos de la cámara.

Normalmente se asume $s=0$, $r=1$ e incluso $o_x=o_y=0$, con lo que pueden Quedar reducidos a un sólo *gdl* (f).

Modelo lineal de cámara (IV)

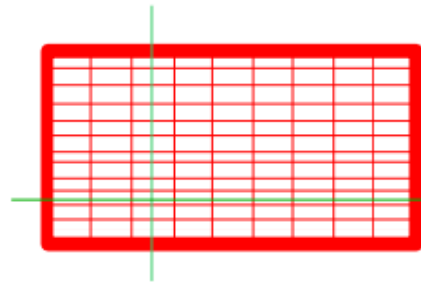
- Parámetros intrínsecos (cont.):
 - Simplificación progresiva de modelos de cámara:



$$\begin{bmatrix} f & s & o_x \\ 0 & fr & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

5 gdl

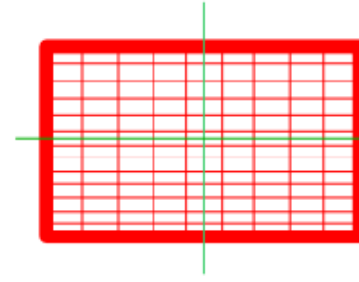
Cámara
general



$$\begin{bmatrix} f & 0 & o_x \\ 0 & fr & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

4 gdl

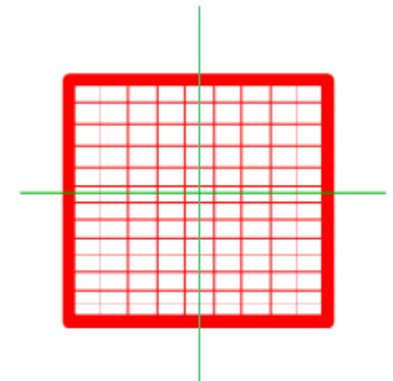
Píxel
rectangular



$$\begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2 gdl

Punto principal
centrado



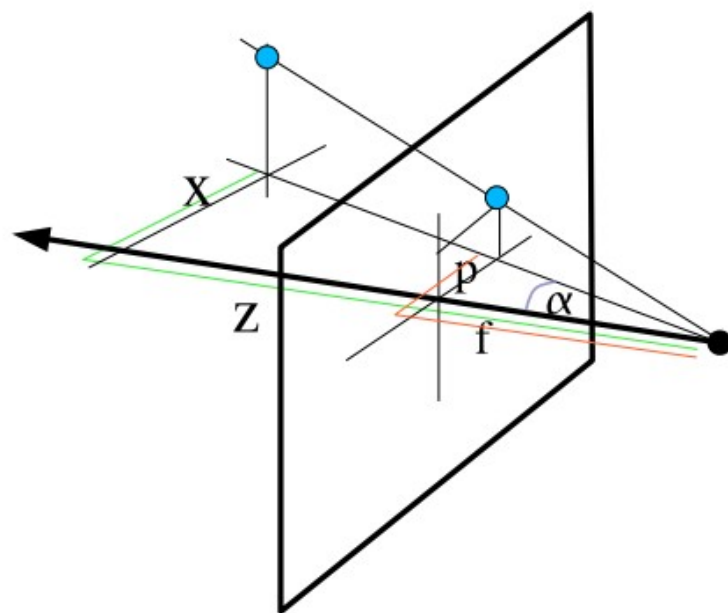
$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1 gdl

Píxel cuadrado,
focal única,
pp centrado

Cámara como medidor de ángulos

- Usando la inversa de la matriz K:



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

$$\frac{x}{z} = \frac{p}{f} = \tan \alpha$$

- Esto permite hacernos una idea del valor aproximado de f midiendo el tamaño en píxeles para un objeto de un tamaño angular conocido (p.e. la Luna = 0.54° vista desde la Tierra).

Matriz de cámara

- Se obtiene combinando los parámetros extrínsecos dados por R y C (6 *gdl*) y los intrínsecos dados por K (5 *gdl*):

$$M = \underset{3 \times 4}{K} \underset{3 \times 3}{R} \left[\underset{3 \times 3}{I} \mid \overset{3 \times 1}{-C} \right] = \underset{3 \times 3}{K} \left[\underset{3 \times 3}{R} \mid \underset{3 \times 1}{t} \right] \quad \begin{matrix} t = -RC \\ 3 \times 1 \quad 3 \times 3 \quad 3 \times 1 \end{matrix}$$

- Las coordenadas homogéneas permiten expresar M como un simple producto de matrices.
- M tiene un total de $6+5=11$ *gdl* ($3 \times 4=12$, al que le quitamos 1 por la homogeneidad de la propia M).
- El *espacio nulo* de M coincide con el centro óptico:

$$MC = 0$$

$$C = (C, 1) \quad 4 \times 1$$

(centro óptico homogéneo)

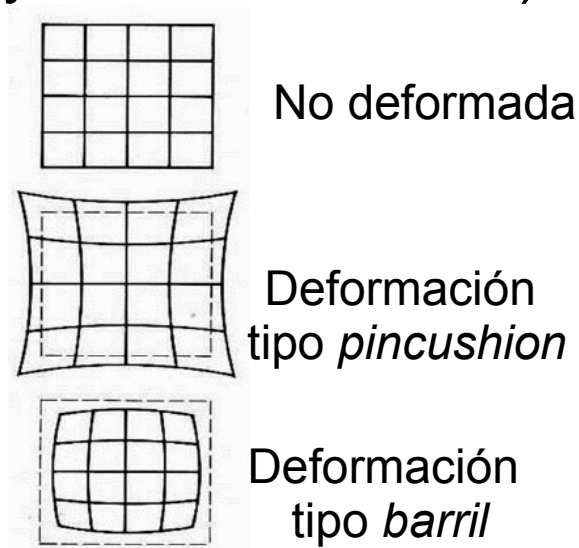
Propiedades de la proyección



- Los puntos se proyectan a puntos, las líneas a líneas.
- NO se preservan los ángulos.
- Las líneas paralelas se cortan en sus *puntos de fuga*.
- La matriz de proyección $P(3 \times 4)$ tiene 11 gdl (no 12) \rightarrow es una magnitud homogénea (escalarla globalmente NO afecta a su comportamiento).

Distorsión radial

- Afecta a la aproximación anterior (las rectas ya no se proyectan como tales):



Especialmente apreciable en objetivos gran angular (ej. *fisheye*)

- Se deshace matemáticamente de forma previa:

Coordenadas sin distorsionar →

$$\begin{aligned} x_u &= x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + \dots) \\ y_u &= y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + \dots) \end{aligned}$$

Coordenadas distorsionadas

Punto principal

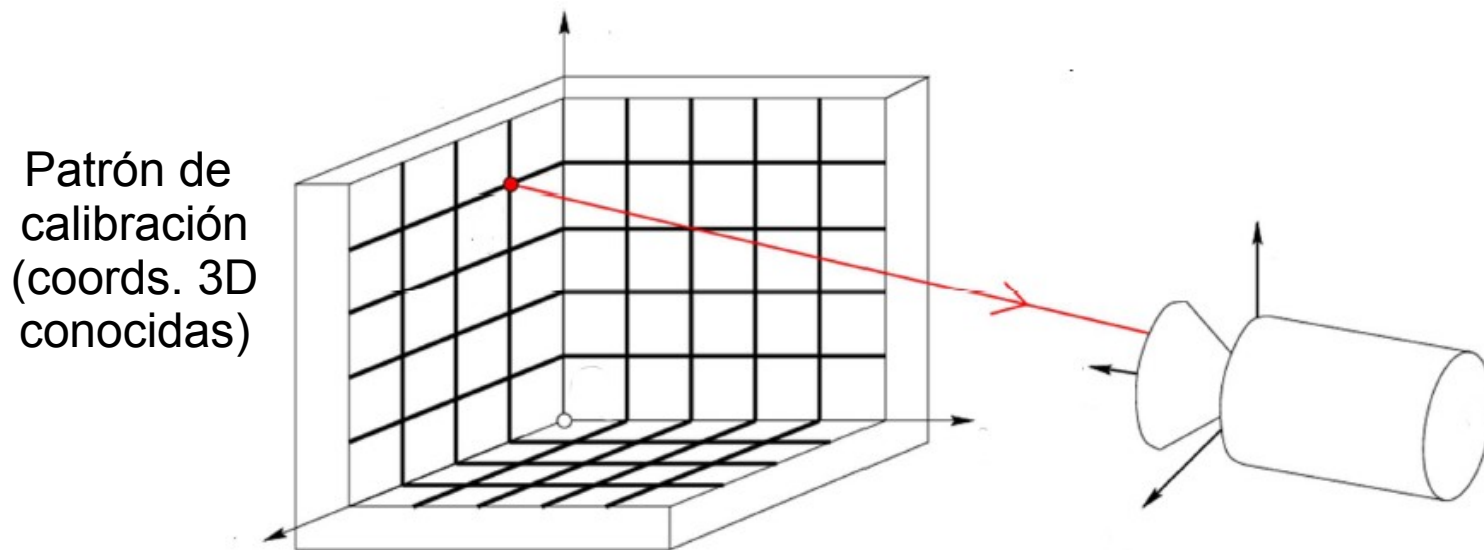
Distancia al punto principal

$$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$$

Coeficientes de distorsión

Calibración (I)

- Problema:
 - Dadas coordenadas de puntos 3D conocidos, y sus correspondientes proyecciones 2D, estimar la matriz M .
 - A partir de M se puede:
 - Realizar realidad aumentada
 - Estimar la posición (*pose*) de la cámara
 - Estimar los parám. intrínsecos de la cámara (calib. intrínseca)

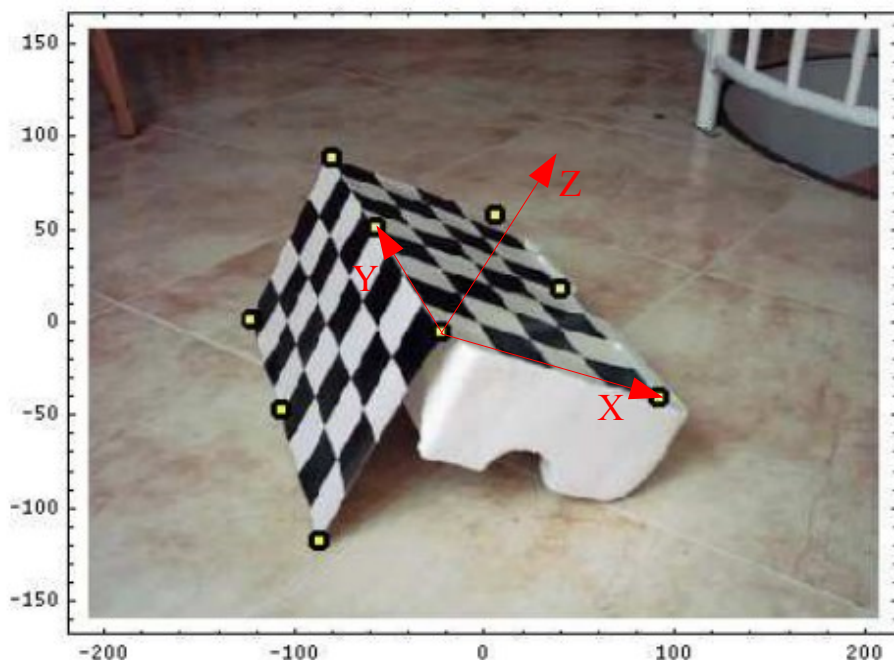


Calibración (II)

- Cada correspondencia $3D \leftrightarrow 2D$ genera 2 ecuaciones:
 - Puesto que hay 11 gdl en $M \rightarrow$ necesarios un mínimo de $5\frac{1}{2}$ (normalmente se usan 6).
 - Se pueden usar más y resolver por mínimos cuadrados.
- Ejemplo:

Coords 3D (unids. = 4x53 mm) Coords 2D (unids. píxel)

$$\begin{pmatrix} x & y & z \\ 0 & 2 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & -1 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} p & q \\ -123 & 2 \\ -107 & -47 \\ -87 & -117 \\ -81 & 89 \\ -57 & 51 \\ -23 & -5 \\ 5 & 58 \\ 40 & 18 \\ 91 & -40 \end{pmatrix}$$



Calibración (III)

- Cada correspondencia $(x,y,z) \leftrightarrow (p,q)$ impone la restricción $M(x,y,z,1) = \lambda(p,q,1)$ (λ , desconocida, viene de la igualdad homogénea).
- Eso implica que $(p,q,1) \times M(x,y,z,1) = \mathbf{0}$ (puesto que ambos vectores han de ser paralelos). Así se elimina λ .
- La igualdad vectorial anterior genera en realidad tres igualdades (de las que en realidad sólo 2 son linealmente independientes, pero suelen usarse las 3 por estabilidad numérica).
- Los coeficientes resultantes para el sistema de ecuaciones lineales homogéneo resultante en $M_{11} \dots M_{34} = m_1 \dots m_{12}$ son:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -x & -y & -z & -1 & qx & qy & qz & q \\ x & y & z & 1 & 0 & 0 & 0 & 0 & -px & -py & -pz & -p \\ -qx & -qy & -qz & -q & px & py & pz & p & 0 & 0 & 0 & 0 \end{bmatrix}$$

Calibración (IV)

- Para nuestro ejemplo, la matriz de coeficientes (A) del **sistema homogéneo completo** (usando las 9 correspondencias disponibles) sería:
 - (Se muestran sólo las líneas correspondientes a las correspondencias 1, 2 y 9):

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & -2 & 1 & -1 & 0 & 4 & -2 & 2 \\
 0 & 2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 246 & -123 & 123 \\
 0 & -4 & 2 & -2 & 0 & -246 & 123 & -123 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & -47 & 47 & -47 \\
 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 107 & -107 & 107 \\
 0 & 47 & -47 & 47 & 0 & -107 & 107 & -107 & 0 & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -40 & 0 & 0 & -40 \\
 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -91 & 0 & 0 & -91 \\
 40 & 0 & 0 & 40 & 91 & 0 & 0 & 91 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 m_1 \\
 m_2 \\
 \vdots \\
 m_{12}
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 \vdots \\
 0
 \end{bmatrix}$$

- Donde los m_i son los 12 coeficientes de la M buscada.

Calibración (V)

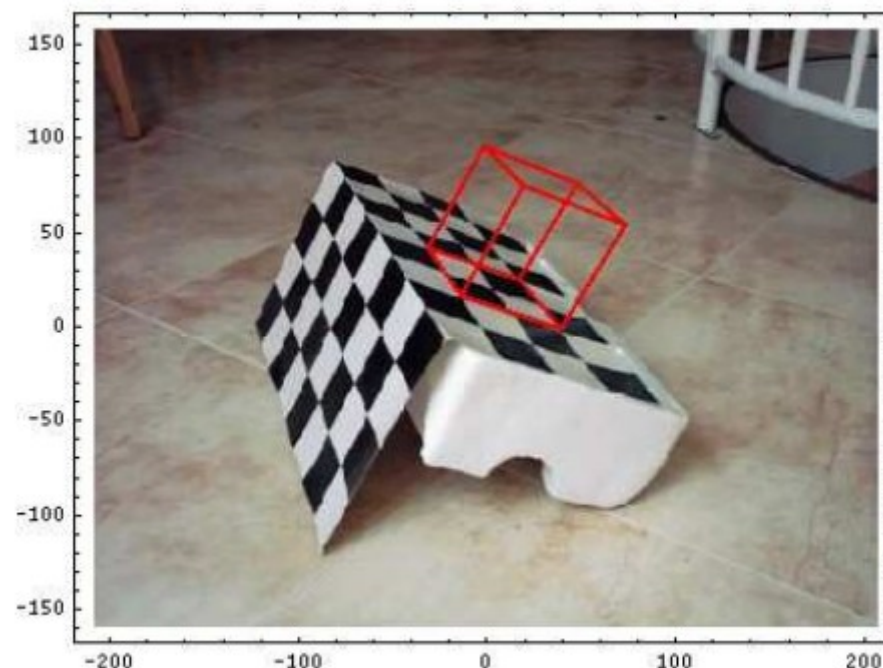
- En este caso, el procedimiento de mínimos cuadrados para resolver $Ax=b$ habitual que usa la pseudoinversa de la matriz de coeficientes ($x=A^+b=(A^TA)^{-1}A^T b$) no funcionaría, puesto que $b=0$ produciría la solución trivial $x=0$.
- En su lugar, se usa como solución el vector propio correspondiente al menor valor propio de la matriz (A^TA) , que es simétrica y definida positiva ...
- ...o, alternativamente, el vector fila derecho correspondiente al menor valor singular de la SVD de la matriz A .
- Ambas soluciones son naturalmente equivalentes.

Calibración (VI)

- En nuestro caso, se obtiene un vector (m_1, \dots, m_{12}) que, convenientemente reescalado para que m_{12} valga 1 (eliminando factor de homogeneidad):

$$M = \begin{pmatrix} 123.088 & -46.516 & 72.955 & -23.079 \\ -39.263 & 66.110 & 124.601 & -4.405 \\ 0.097 & 0.217 & -0.107 & 1 \end{pmatrix}$$

- Con dicha M ya podemos reprojectar objetos virtuales con las coordenadas 3D que se deseen (*realidad aumentada*):



Identificación de parámetros

- Una vez tenemos la matriz de cámara M , podemos factorizarla adecuadamente para obtener su posición (dada por R y C) y su matriz de intrínsecos K .
- Para ello, se usa la **descomposición RQ** , que descompone una matriz cuadrada 3×3 en forma de un producto de una matriz triangular superior R por una rotación Q (¡OJO con la notación cambiada aquí para R !).
- La factorización resultante para la submatriz 3×3 izquierda del ejemplo anterior resulta en las dos matrices K y R siguientes:

$$K = \begin{bmatrix} 569.96 & 24.83 & -86.78 \\ 0 & 560.13 & -40.18 \\ 0 & 0 & 1.00 \end{bmatrix} \quad R = \begin{bmatrix} -0.90 & 0.21 & -0.39 \\ 0.24 & -0.51 & -0.82 \\ -0.37 & -0.83 & 0.41 \end{bmatrix}$$

- Finalmente, C puede obtenerse como el espacio nulo de M (último vector singular derecho de su SVD):

Último vector singular (espacio nulo): Transformación homogénea:
(-0.438291, -0.806473, 0.304019, 0.255098) \rightarrow (-1.71813, -3.16143, 1.19177)

Detalles de implementación

- Lo anterior es el esquema básico del procedimiento...
- ... pero hay una serie de **detalles importantes** que han de tratarse para obtener un algoritmo robusto:
 - **Normalización de puntos 2D/3D de entrada**
 - Para evitar problemas numéricos derivados de la presencia de valores de magnitudes muy diferentes en la matriz de coeficientes A.
 - **Detalles de la descomposición RQ**
 - La descomposición no es única (puede haber cambios de signo) → compensar.
 - Hay que forzar a que K (R en descomposición RQ) sea de diagonal positiva y con 1 en K_{33} .
 - Finalmente, si la rotación Q queda con determinante -1 cambiamos de signo para que sea una rotación sin "reflexión".

Normalización (I)

- Se pretransforman tanto los puntos de entrada 2D como los 3D con las siguientes matrices:

$$T_{2D} = \begin{pmatrix} 1/scale_{2D} & 0 & 0 \\ 0 & 1/scale_{2D} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -mean_x \\ 0 & 1 & -mean_y \\ 0 & 0 & 1 \end{pmatrix}$$

Distancia media a los centroides 2D/3D

Centroide de los puntos 2D/3D

$$T_{3D} = \begin{pmatrix} 1/scale_{3D} & 0 & 0 & 0 \\ 0 & 1/scale_{3D} & 0 & 0 \\ 0 & 0 & 1/scale_{3D} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -mean_X \\ 0 & 1 & 0 & -mean_Y \\ 0 & 0 & 1 & -mean_Z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

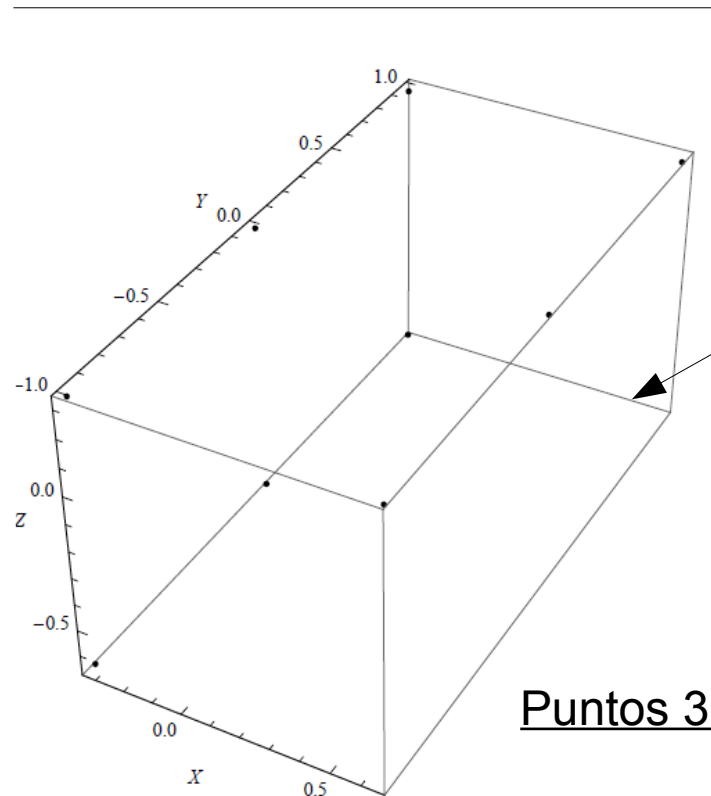
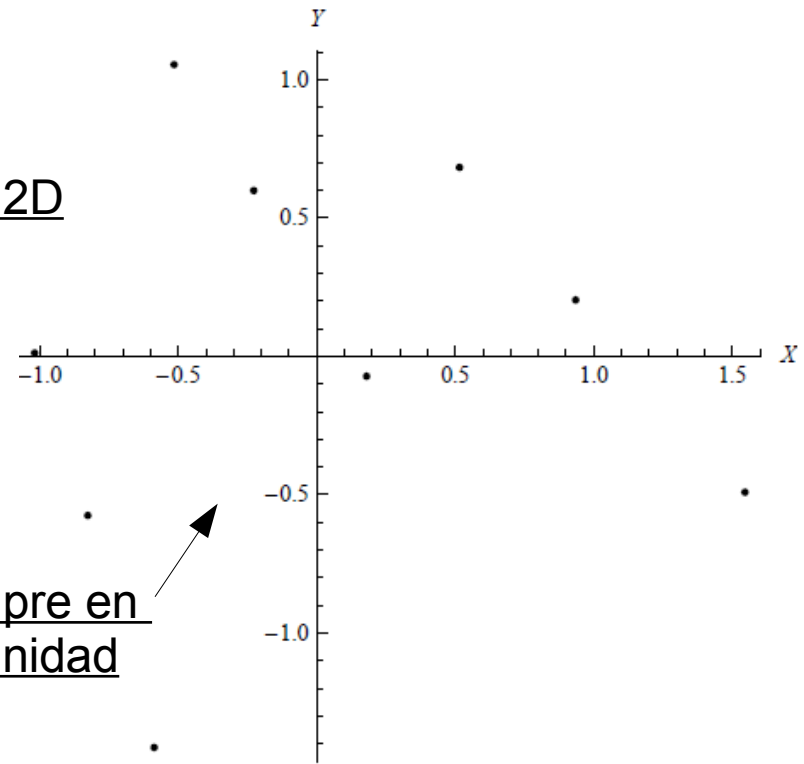
- Con esto conseguimos centrarlos en torno al origen, y darles distancias a éste en torno a la unidad.

Normalización (II)

- En nuestro caso, quedan las siguientes matrices y puntos 2D/3D transformados:

$$T_{2D} = \begin{pmatrix} 0.0119839 & 0. & 0.455389 \\ 0. & 0.0119839 & -0.0119839 \\ 0. & 0. & 1. \end{pmatrix}$$

Puntos 2D



Valores siempre en torno a la unidad

Puntos 3D

$$T_{3D} = \begin{pmatrix} 0.98231 & 0. & 0. & -0.327437 \\ 0. & 0.98231 & 0. & -0.98231 \\ 0. & 0. & 0.98231 & 0.327437 \\ 0. & 0. & 0. & 1. \end{pmatrix}$$

Normalización (III)


- Usando las coordenadas normalizadas resultantes, la matriz A resultante tiene (por construcción) valores en torno a la unidad en valor absoluto → **mejor condicionamiento numérico**.
- La nueva matriz M' estimada, de todos modos, obviamente será distinta, puesto que mapea de coordenadas 3D a 2D normalizadas:

$$M' = \begin{pmatrix} 0.53875 & -0.162184 & 0.293538 & -0.0170632 \\ -0.164671 & 0.281249 & 0.529741 & 0.0242336 \\ 0.0359944 & 0.0773735 & -0.0395086 & 0.448012 \end{pmatrix}$$

Normalización (IV)

- Por ello, es necesaria una “denormalización” final:

$$\begin{aligned}
 M &= T_{2D}^{-1} \cdot M \cdot T_{3D} \\
 &= \begin{pmatrix} 83.4452 & 0. & -38. & \\ 0. & 83.4452 & 1. & \\ 0. & 0. & 1. & \end{pmatrix} \cdot \begin{pmatrix} 0.53875 & -0.162184 & 0.293538 & -0.0170632 \\ -0.164671 & 0.281249 & 0.529741 & 0.0242336 \\ 0.0359944 & 0.0773735 & -0.0395086 & 0.448012 \end{pmatrix} \cdot \\
 &\quad \cdot \begin{pmatrix} 0.98231 & 0. & 0. & -0.327437 \\ 0. & 0.98231 & 0. & -0.98231 \\ 0. & 0. & 0.98231 & 0.327437 \\ 0. & 0. & 0. & 1. \end{pmatrix} \\
 &\quad \approx \begin{pmatrix} 123.291 & -46.5966 & 73.53 & -23.1121 \\ -38.7652 & 66.6017 & 124.922 & -4.92633 \\ 0.101812 & 0.218854 & -0.111752 & 1. \end{pmatrix}
 \end{aligned}$$

Igualdad homogénea 

- La M resultante es lógicamente similar a la anterior, pero está calculada de forma mucho más segura (robusta numéricamente).

Descomposición RQ (I)

- Ojo, no es lo mismo QR (más habitual) que RQ (la que aquí necesitamos):
 - Si RQ no disponible, puede calcularse así:

```
function [R,Q] = rq(M)
    [Q,R] = qr(flipud(flipplr(M))');
    R = flipplr(flipud(R'));
    Q = flipplr(flipud(Q'));
end
```


Descomposición RQ (II)

- Para obtener las matrices de intrínsecos K y de rotación R hay que (en detalle):
 1. Obtener la descomposición RQ de la submatriz 3x3 izquierda de M .
 2. Reescalar K (= R de la RQ, triangular superior) para que el elemento (3,3) sea 1.
 3. Si obtenemos valores negativos en la diagonal de K tenemos que cambiar de signo la columna de K y la fila de Q (= la otra R en nuestro caso, e.d., la rotación de la cámara).
 4. Finalmente, si Q quedaba con determinante -1 cambiamos de signo todos sus elementos (para obtener una rotación sin "reflexión").

Comprobación de resultados (I)

- Primero, podemos comprobar que los valores obtenidos para K, R y C de la descomposición de M la reconstruyen perfectamente:

$$R = \begin{pmatrix} -0.894165 & 0.215593 & -0.392415 \\ 0.232296 & -0.525871 & -0.818229 \\ -0.382764 & -0.822788 & 0.420134 \end{pmatrix} \quad C = \begin{pmatrix} -1.71813 \\ -3.16143 \\ 1.19177 \end{pmatrix}$$

$$K = \begin{pmatrix} 560.706 & 26.3934 & -82.8601 \\ 0. & 549.807 & -47.0804 \\ 0. & 0. & 1. \end{pmatrix}$$

Igualdad
homogénea

$$M = KR[I] - C = K[R|t] \approx \begin{pmatrix} 123.291 & -46.5966 & 73.53 & -23.1121 \\ -38.7652 & 66.6017 & 124.922 & -4.92633 \\ 0.101812 & 0.218854 & -0.111752 & 1. \end{pmatrix}$$

- El hecho de que $K_{11} \simeq K_{22}$, y $K_{12} \simeq 0$ (en proporción) nos dice que el píxel es aproximadamente cuadrado \rightarrow “buena señal”
- El centro óptico puede salir más o menos desplazado del centro (es más inestable), pero en principio no debería “salirse” de la imagen.

Comprobación de resultados (II)

- Del valor de C podemos comprobar, más o menos “a ojo”, que se corresponde con la posición de la cámara (en el mismo sistema de coordenadas que los puntos 3D).
- Finalmente, debemos comprobar que la M así obtenida proyecta correctamente cada punto 3D a su proyección 2D asociada.
 - Por ejemplo, para la correspondencia 3D \leftrightarrow 2D primera, es decir $(0,2,-1) \leftrightarrow (-123,2)$:

$$\begin{pmatrix} 123.291 & -46.5966 & 73.53 & -23.1121 \\ -38.7652 & 66.6017 & 124.922 & -4.92633 \\ 0.101812 & 0.218854 & -0.111752 & 1. \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 2 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -189.835 \\ 3.35477 \\ 1.54946 \end{pmatrix} \begin{matrix} \approx \\ \nearrow \text{Igualdad} \\ \text{homogénea} \end{matrix} \begin{pmatrix} -122.517 \\ 2.16512 \\ 1 \end{pmatrix} \approx \begin{matrix} \nearrow \text{Aproximadamente igual} \\ \text{(error de reproyección inferior a un píxel)} \end{matrix} \begin{pmatrix} -123 \\ 2 \\ 1 \end{pmatrix}$$

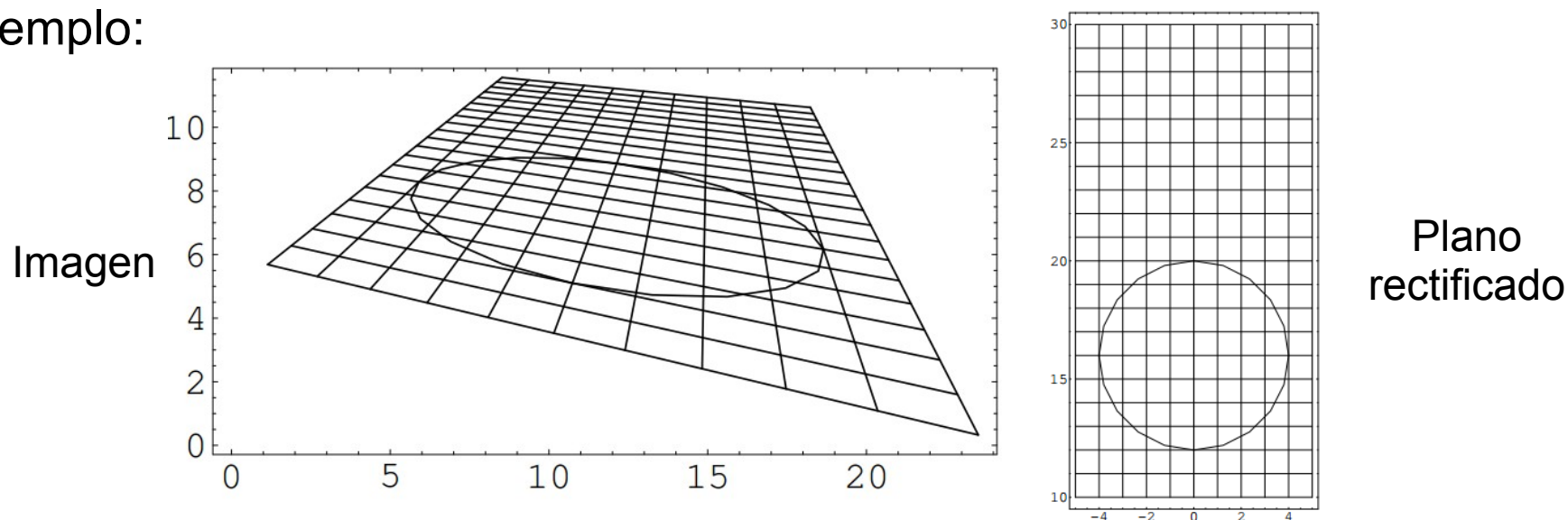
Aproximadamente igual
(error de reproyección inferior a un píxel)

Rectificación de planos (I)

- La transformación $3D \leftrightarrow 2D$ (proyección escena \rightarrow cámara) es no invertible:
 - No podemos extraer la “profundidad” de un punto 3D proyectado cualquiera.
- Sin embargo, la correspondencia entre planos es biyectiva (uno a uno):
 - \rightarrow Podemos “deshacer” la transformación perspectiva de un plano (decimos que **lo rectificamos**).

Rectificación de planos (II)

- La transformación se modela con una matriz H 3x3, denominada “homografía planar”.
- Ejemplo:



- Si conocemos la K , y la C y la R respecto al sistema de coordenadas del mundo, podemos obtener la H a partir de M (p.e., simplemente eliminando la tercera columna de M , si el plano coincide con el plano XY del espacio)...
- ... pero, al tratarse de una homografía (matriz 3x3 \rightarrow 8 *gdl*), con sólo 4 correspondencias de puntos (cada una impone dos restricciones linealmente independientes sobre H) se puede computar la misma.

Estimación de la matriz H (I)

- Se hace a partir de 4 puntos, de forma completamente análoga a como reseccionábamos la cámara:
- Cada correspondencia $2D \leftrightarrow 2D$ impone 2 restricciones del tipo $(p, q, 1) \times H(x, y, 1) = (0, 0, 0) \rightarrow 3$ restricciones homogéneas (sólo 2 linealmente independientes) con filas de coeficientes (matriz A):

$$\begin{bmatrix} 0 & 0 & 0 & -x & -y & -1 & qx & qy & q \\ x & y & 1 & 0 & 0 & 0 & -px & -py & -p \\ -qx & -qy & -q & px & py & p & 0 & 0 & 0 \end{bmatrix}$$

- Por tanto, con 4 de estas correspondencias “atamos” los 8 *gdl* de una matriz H de $3 \times 3 = 9$ elementos.
- Se podría hacer también de forma análoga usando 4 rectas en lugar de 4 puntos (también en coordenadas homogéneas).

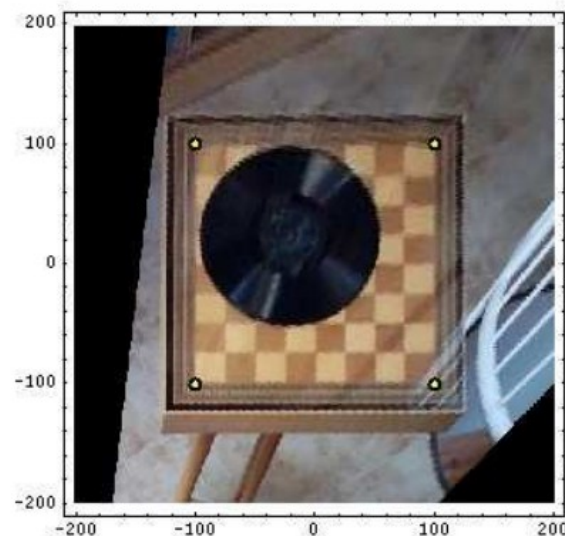
Estimación de la matriz H (II)

- Todo lo aplicado a la resección de cámaras es también válido aquí:
 1. Puede estimarse H con más de 4 correspondencias, usando mínimos cuadrados.
 2. Se resuelve mediante el vector de menor valor propio de la matriz $(A^T A)$, o, alternativamente, el vector fila derecho correspondiente al menor valor singular de la SVD de la matriz A .
 3. Es conveniente prenormalizar las entradas para evitar problemas numéricos (\rightarrow luego hay que pre- y post-multiplicar la H estimada por las correspondientes matrices de transformación):

$$H = T'_{2D}{}^{-1} \cdot H_{norm} \cdot T''_{2D}$$

Ejemplo de estimación de H

- Rectificación **frontoparalela** de un tablero de ajedrez:



- Se muestran las coordenadas originales, las deseadas, y la H correspondiente obtenida (naturalmente, H^{-1} lleva de la imagen rectificada de nuevo a la original):

$$\begin{pmatrix} x & y \\ -171 & 109 \\ -120 & 31 \\ 117 & 53 \\ 11 & 115 \end{pmatrix} \rightarrow \begin{pmatrix} x' & y' \\ -100 & 100 \\ -100 & -100 \\ 100 & -100 \\ 100 & 100 \end{pmatrix} \quad H = \begin{bmatrix} 0.63868 & 0.77290 & -39.73059 \\ -0.11082 & 1.94177 & -165.90578 \\ -0.00034 & -0.00378 & 1 \end{bmatrix}$$

- Ejercicio \rightarrow intentar deducir (p.e. con Numpy/Matlab) cómo se ha obtenido H

Homografía de un plano 3D

- Una matriz de cámara $M = K[R|t]$ “contiene” las homografías entre cualquier plano del mundo y la imagen.
- Por ejemplo, dado el **plano $Z=0$** , la homografía inducida tiene la expresión:

$$H = K[\hat{R}|t] = K[r_1|r_2|t]$$

- donde K es la matriz de intrínsecos, y \hat{R} es una matriz de rotación sin la tercera columna (que se hace “invisible” porque todos los puntos del plano tienen la coordenada $Z=0$), mientras que $t=-RC$ es un vector arbitrario.
- \hat{R} consiste en dos vectores columna perpendiculares y de longitud unidad (r_1 y r_2).

Homografía de una rotación pura (I)

- Dos imágenes tomadas desde el mismo lugar (centro de proyección común, el resto de parámetros pueden variar) están relacionadas mediante una homografía plana.
- Supongamos que estamos en el origen y tenemos dos cámaras $M_1 = K[I|0]$ y $M_2 = K[R|0]$.
- Un punto del mundo $X = (X, Y, Z, W)^T$ se ve respectivamente como $x = K(X, Y, Z)^T$ y $x' = KR(X, Y, Z)^T$.
- Por tanto, $x' = KRK^{-1}x$, con lo que ambas vistas están relacionadas por una homografía con la estructura $H = KRK^{-1}$.

Homografía de una rotación pura (II)

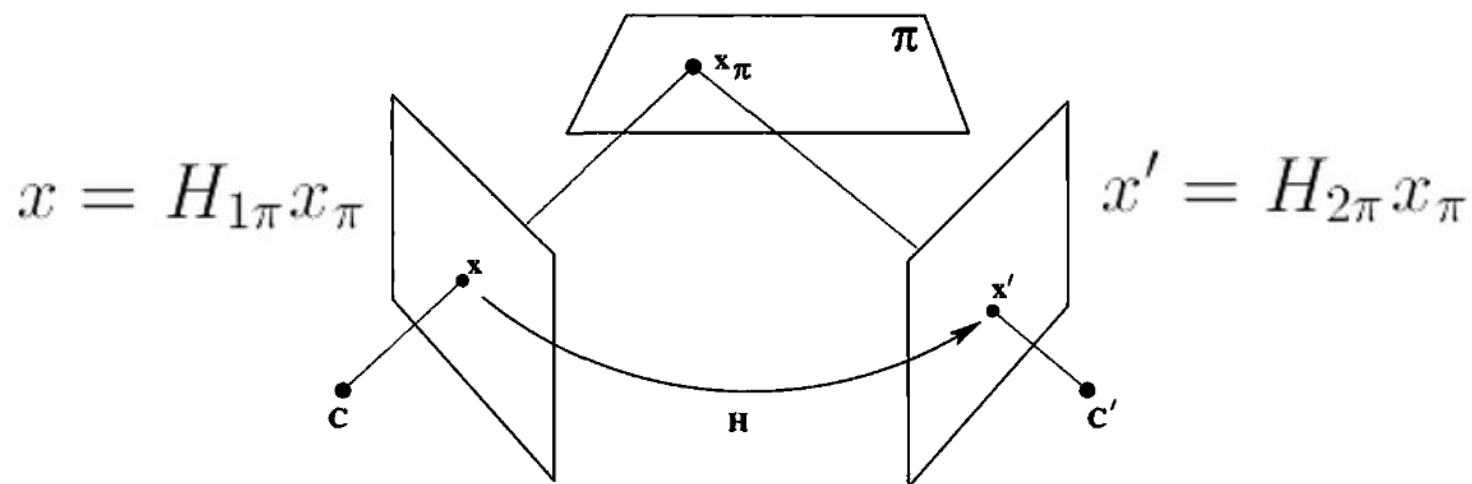
- **Mosaicos de imágenes:**

- Lo anterior permite la fabricación sencilla de mosaicos, transformando varias vistas de una escena tomadas desde el mismo punto (pueden variar la orientación, zoom, etc.) a un sistema de referencia común:
- Ejemplo de panorámica con tres imágenes del campus, los puntos marcados (4 comunes a cada par) se usaron en la estimación de las H de transferencia de las imágenes laterales al marco común central):



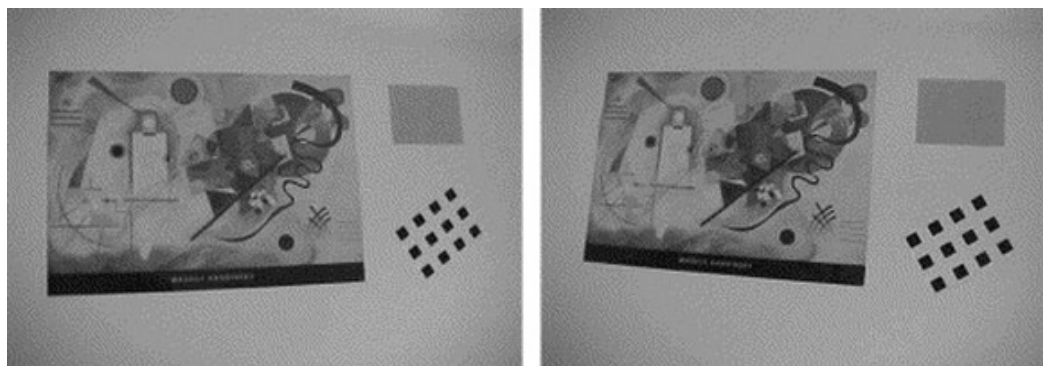
Homografías interimagen

- La visión desde dos puntos distintos de un mismo plano 3D induce una llamada homografía interimagen:



$$H = H_{2\pi} H_{1\pi}^{-1}$$

$$x' = H_{2\pi} H_{1\pi}^{-1} x = Hx$$



Tratamiento de rectas en la imagen (I)

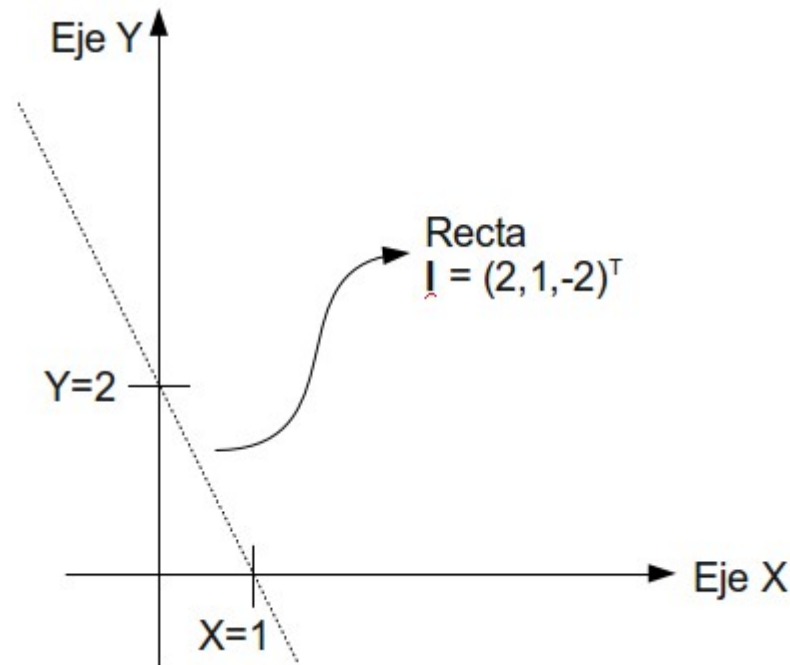
- Al igual que los puntos 2D, las rectas 2D serán tratadas como vectores homogéneos:

$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \longrightarrow \mathbf{l}^T \cdot \mathbf{x} = 0 \Leftrightarrow ax + by + c = 0$$

Coordenadas homogéneas de la recta

Ecuación que cumplen los puntos $\mathbf{x} = (x, y, 1)$ de la recta

Ejemplo:

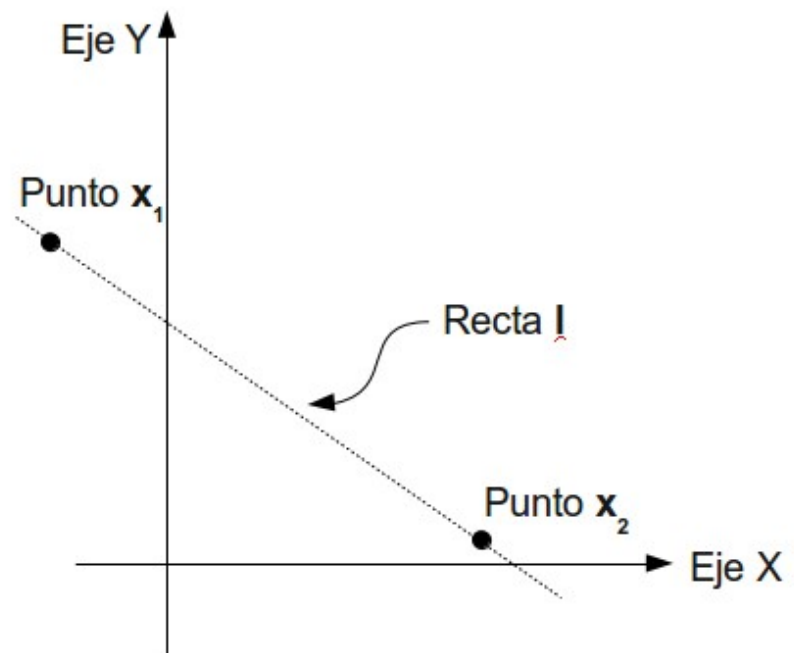
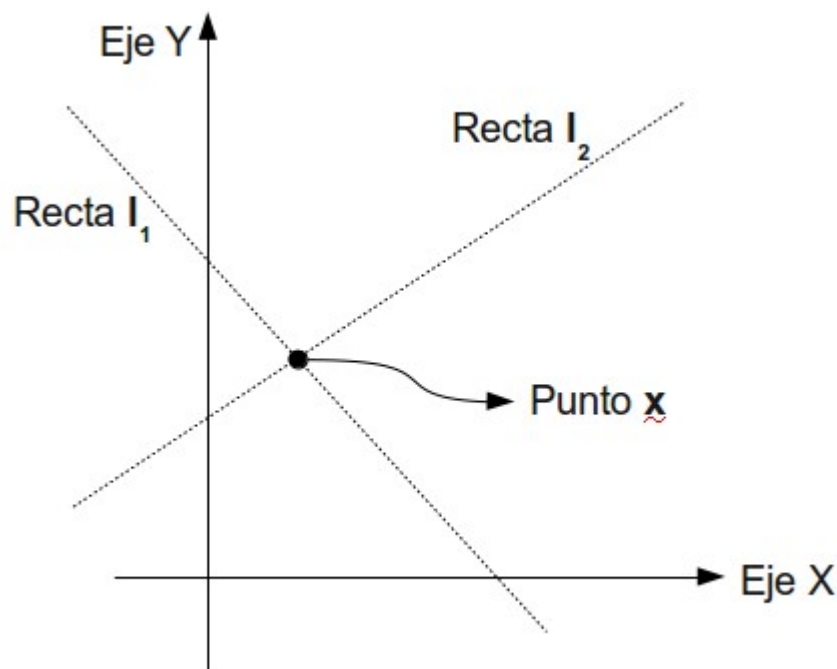


Tratamiento de rectas en la imagen (II)

- Las propiedades más interesantes de esta representación son la facilidad para determinar:

- El punto \mathbf{x} de intersección de dos rectas \mathbf{l}_1 y \mathbf{l}_2 : $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$

- La recta \mathbf{l} que pasa por dos puntos \mathbf{x}_1 y \mathbf{x}_2 : $\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$



Tratamiento de rectas en la imagen (III)

- Recordar que el **producto vectorial** se computa así:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix} = \begin{vmatrix} u_y & u_z \\ v_y & v_z \end{vmatrix} \cdot \mathbf{i} - \begin{vmatrix} u_x & u_z \\ v_x & v_z \end{vmatrix} \cdot \mathbf{j} + \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix} \cdot \mathbf{k}$$

es decir,

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

- Aunque a menudo se expresa como un producto matriz vector, así:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Notación muy empleada:
matriz “*cross product*”

Tratamiento de rectas en la imagen (IV)

- Otra ventaja de esta representación es que no se necesita ningún tratamiento especial para los **puntos y las líneas en el infinito**:
 - Los puntos en el infinito tienen la forma $\mathbf{x} = (x, y, 0)$.
 - La línea del infinito tiene la forma $\mathbf{l} = (0, 0, 1)$.
 - Las líneas paralelas se cortan en un punto en el infinito.
 - La línea que pasa por un punto en una dirección dada es la que une dicho punto con el punto en el infinito que hay en dicha dirección.
 - Etc...

Tratamiento de rectas en la imagen (V)

- Finalmente, dada una homografía H que transfiere puntos $\mathbf{x} = (x, y, 1)$ según la ecuación:

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \quad \longleftrightarrow \quad \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

- Transfiere también líneas según la siguiente ecuación (es fácil razonar el porqué, algebraicamente):

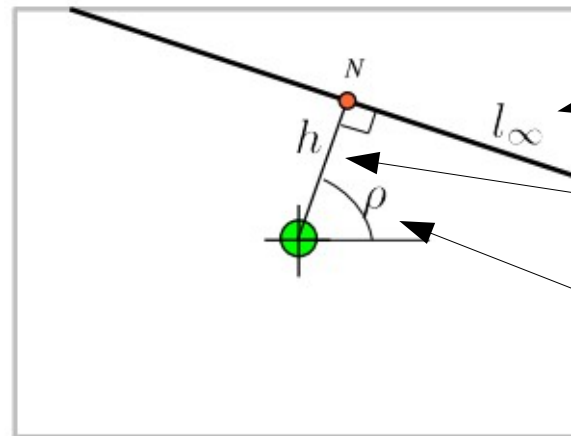
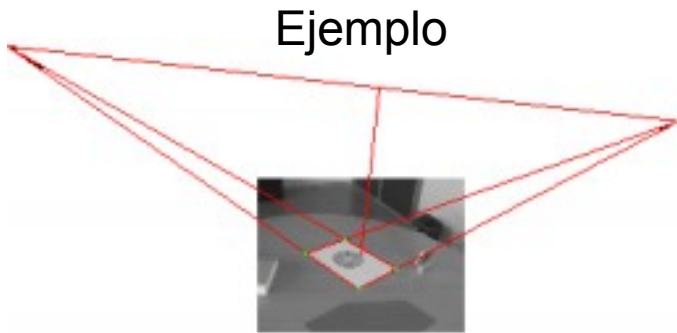
$$\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$$

Calibración intrínseca

- Con todo lo anterior se pueden proponer un par de métodos de **calibración intrínseca de la cámara**:
 1. Un método basado en la localización de dos **puntos de fuga que en el mundo real formen un ángulo recto**.
 2. Otro método basado en la peculiar forma algebraica de la **homografía plano 3D-imagen**.
- **Nota:** se trata de dos métodos sencillos para ilustrar la *autocalibración*, pero por supuesto no son los únicos, ni siquiera los más recomendables.

Calibración mediante puntos de fuga de un ángulo recto

- Supongamos que podemos identificar la línea del horizonte (l_∞) de un plano:

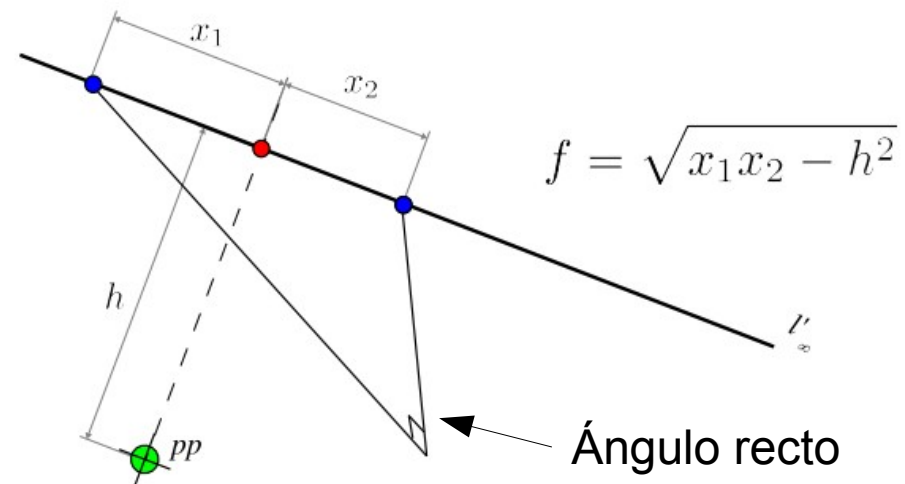


Línea del horizonte

Distancia al punto principal
(altura del horizonte)

Ángulo de *roll*

- Si entonces identificamos un ángulo recto en el plano, y $K = \text{diag}(f, f, 1)$, se cumple la siguiente fórmula, que se puede emplear para calibrar:



Calibración basada en homografía plano 3D – imagen (I)

- Sabemos que la homografía del plano $Z=0$ tiene esta forma (transp. 43):

$$H = K[\hat{R}|t] = K[r_1|r_2|t]$$

- Entonces, si definimos $W=K^{-T} \cdot K^{-1}$, y por la propiedad de las matrices ortogonales, $R \cdot R^T = I$, se cumplirá que:

$$W := K^{-T} K^{-1} \Rightarrow \mathbf{H}^T \cdot W \cdot \mathbf{H} = \begin{bmatrix} r_1 \\ r_2 \\ t \end{bmatrix} K^T K^{-T} K^{-1} K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} =$$

$$\begin{bmatrix} r_1 \\ r_2 \\ t \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} = \begin{bmatrix} 1 & 0 & . \\ 0 & 1 & . \\ . & . & . \end{bmatrix} \approx \begin{bmatrix} a & 0 & b \\ 0 & a & c \\ b & c & d \end{bmatrix}$$

Calibración basada en homografía plano 3D – imagen (II)

- Lo anterior fuerza dos condiciones sobre la matriz $H^T \cdot W \cdot H$:
 1. Que los elementos (1,1) y (2,2) de la misma han de ser iguales.
 2. Que el elemento (1,2) ha de ser nulo –y, como es simétrica, también lo cumplirá el elemento (2,1)–.
- Si suponemos $K = \text{diag}(f, f, 1)$, entonces $W = \text{diag}(g, g, 1) = \text{diag}(1/f^2, 1/f^2, 1)$, y por tanto podemos generar sendas ecuaciones con una sólo incógnita (g) partiendo de los elementos de $H \rightarrow$ *dos alternativas válidas para calibrar*.
- Conocida g , es ya inmediato obtener la focal f , y por tanto calibrar la cámara a partir de cualquier H (homografía plano 3D – imagen).

Bibliografía básica

- "*Multiple view geometry*" (2nd ed.), R. Hartley, A. Zisserman (2003), Cambridge University Press.
- "*Apuntes de Sistemas de Percepción y Visión por Computador*":
(<http://dis.um.es/profesores/alberto/material/percep.pdf>)
- Documentación SciPy / NumPy:
(<http://docs.scipy.org/doc/>)