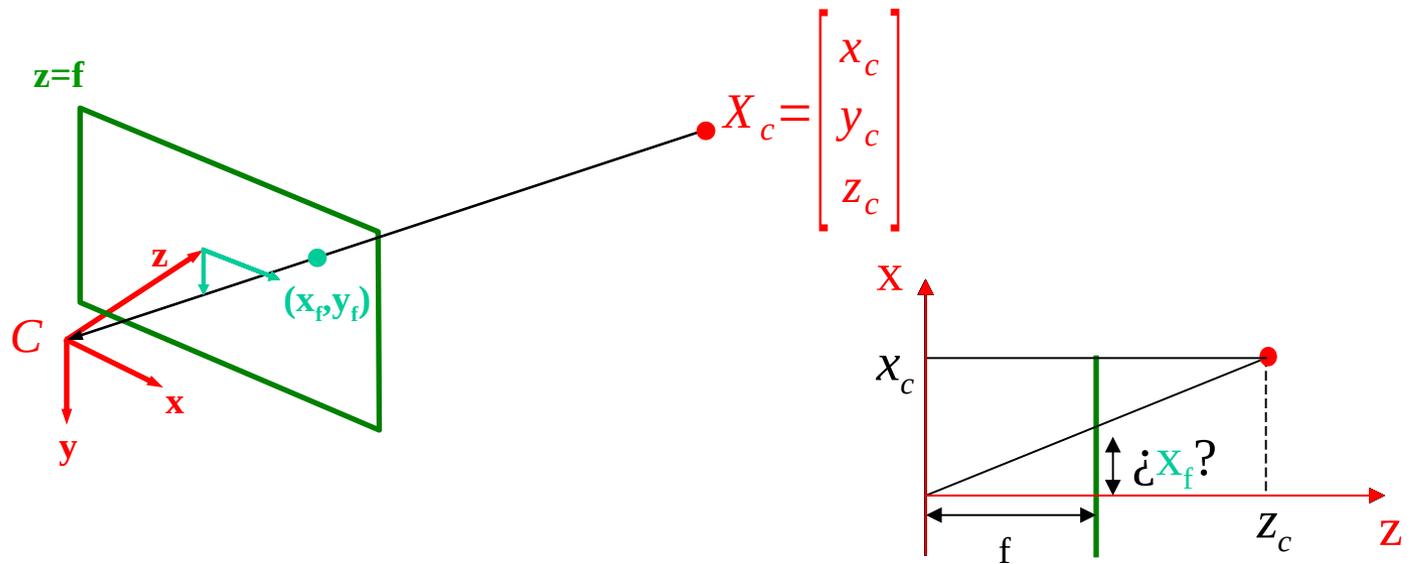


# Geometría de una y dos vistas

# Motivación

- Contar con una cámara calibrada permite
  - Corrección de distorsión
  - Realidad Aumentada
- Conocer la geometría de dos vistas
  - Reconstrucción de una escena

# Modelo pinhole

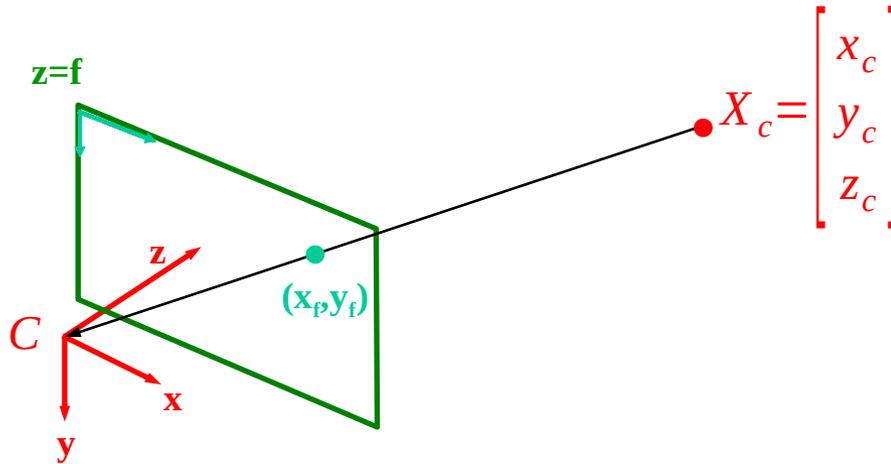


$$x_f = \frac{x_c}{z_c} f$$

$$y_f = \frac{y_c}{z_c} f$$

$$\begin{bmatrix} f x_c \\ f y_c \\ z_c \end{bmatrix} = \begin{bmatrix} f \\ f \\ 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

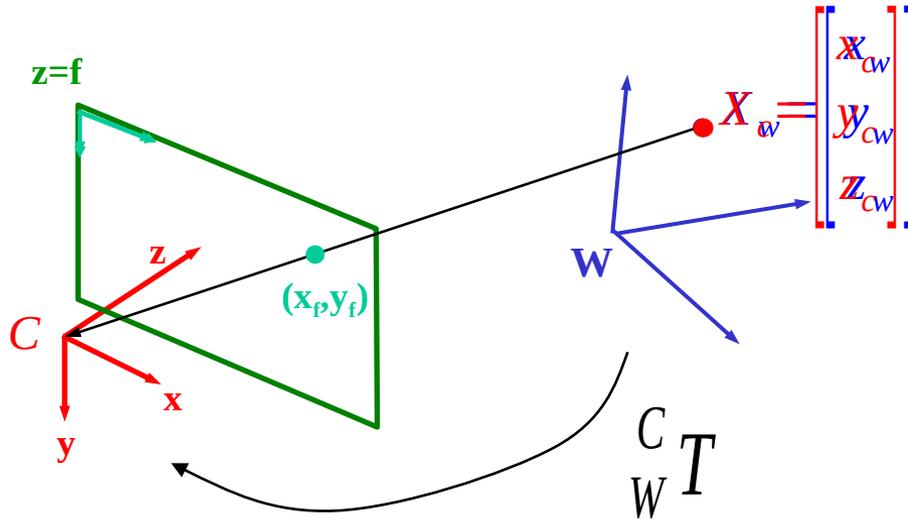
# Modelo pinhole



Cambio de sistema de coordenadas

$$\begin{aligned}
 x_f &= \frac{x_c}{z_c} f + p_x \\
 y_f &= \frac{y_c}{z_c} f + p_y
 \end{aligned}
 \quad
 \begin{bmatrix}
 fx_c + p_x z_c \\
 fy_c + p_y z_c \\
 z_c
 \end{bmatrix}
 =
 \begin{bmatrix}
 f & p_x & 0 \\
 f & p_y & 0 \\
 f & 1 & 0
 \end{bmatrix}
 \begin{bmatrix}
 x_c \\
 y_c \\
 z_c \\
 1
 \end{bmatrix}$$

# Modelo pinhole



$$X_C = {}^C_W T X_W$$

$$X_C = R(X_W - C_W)$$

$$X_C = [R|t] X_W \quad t = -RC_W$$

# Modelo pinhole

- Parámetros extrínsecos  $[R|t]$

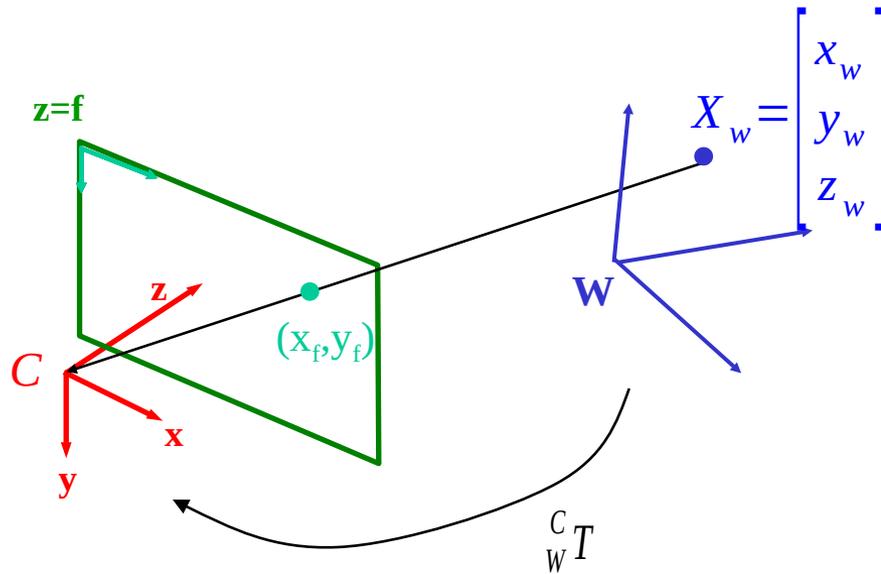
- Parámetros intrínsecos  $K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$

- píxeles no cuadrados  $\alpha_x = \frac{f}{\text{ancho pixel en x}}$   $\alpha_y = \frac{f}{\text{ancho pixel en y}}$   
 $u_0 = \frac{p_x}{\text{ancho pixel en x}}$   $v_0 = \frac{p_y}{\text{ancho pixel en y}}$

- ejes no perpendiculares

$$K = \begin{bmatrix} \alpha_x & s & u_0 \\ & \alpha_y & v_0 \\ & & 1 \end{bmatrix}$$

# Modelo pinhole



$$s \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\tilde{m}} = \underbrace{\begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}}_{[R \ t]} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\tilde{X}_w}$$

# Matriz de proyección

$$P = \underbrace{\begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}}_{[R \ t]}$$

- P tiene 11 grados de libertad

# Calibración de cámara

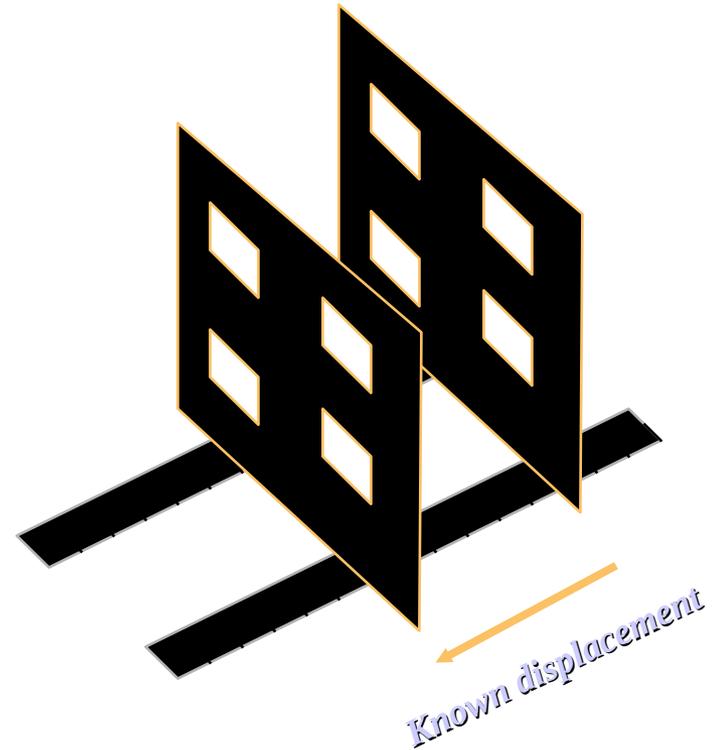
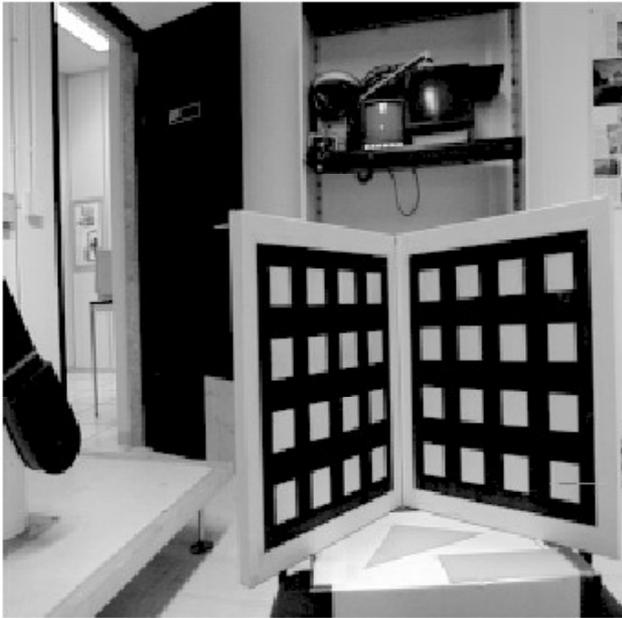
# Calibración de cámara

- Consiste en determinar las características de la cámara (largo focal, relación de aspecto, punto principal)
- Se busca minimizar el error geométrico:

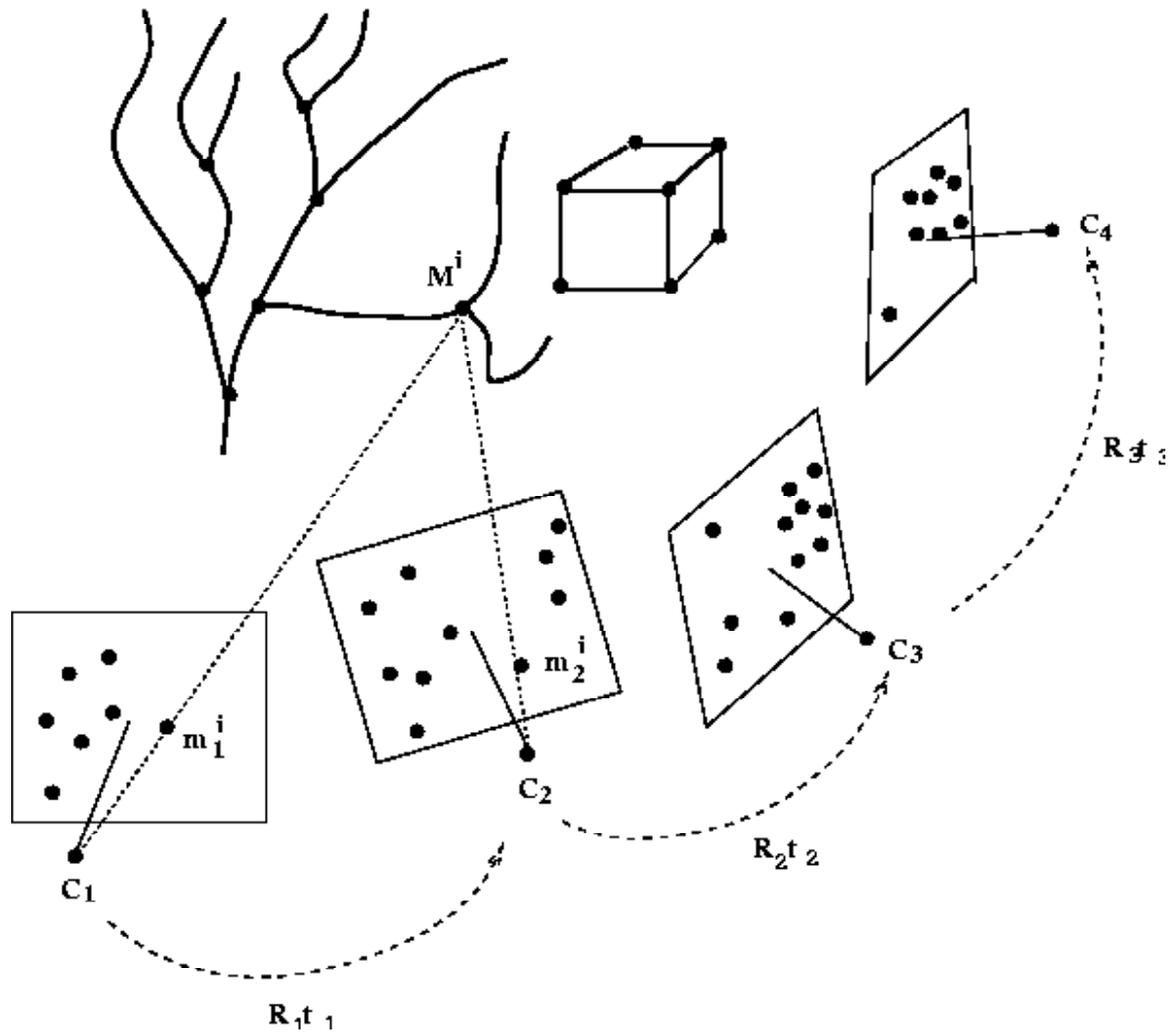
$$\min_{\mathbf{P}} \sum_i d(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i)^2$$

# Enfoque clásico

- Utilizar posiciones 3D muy confiables

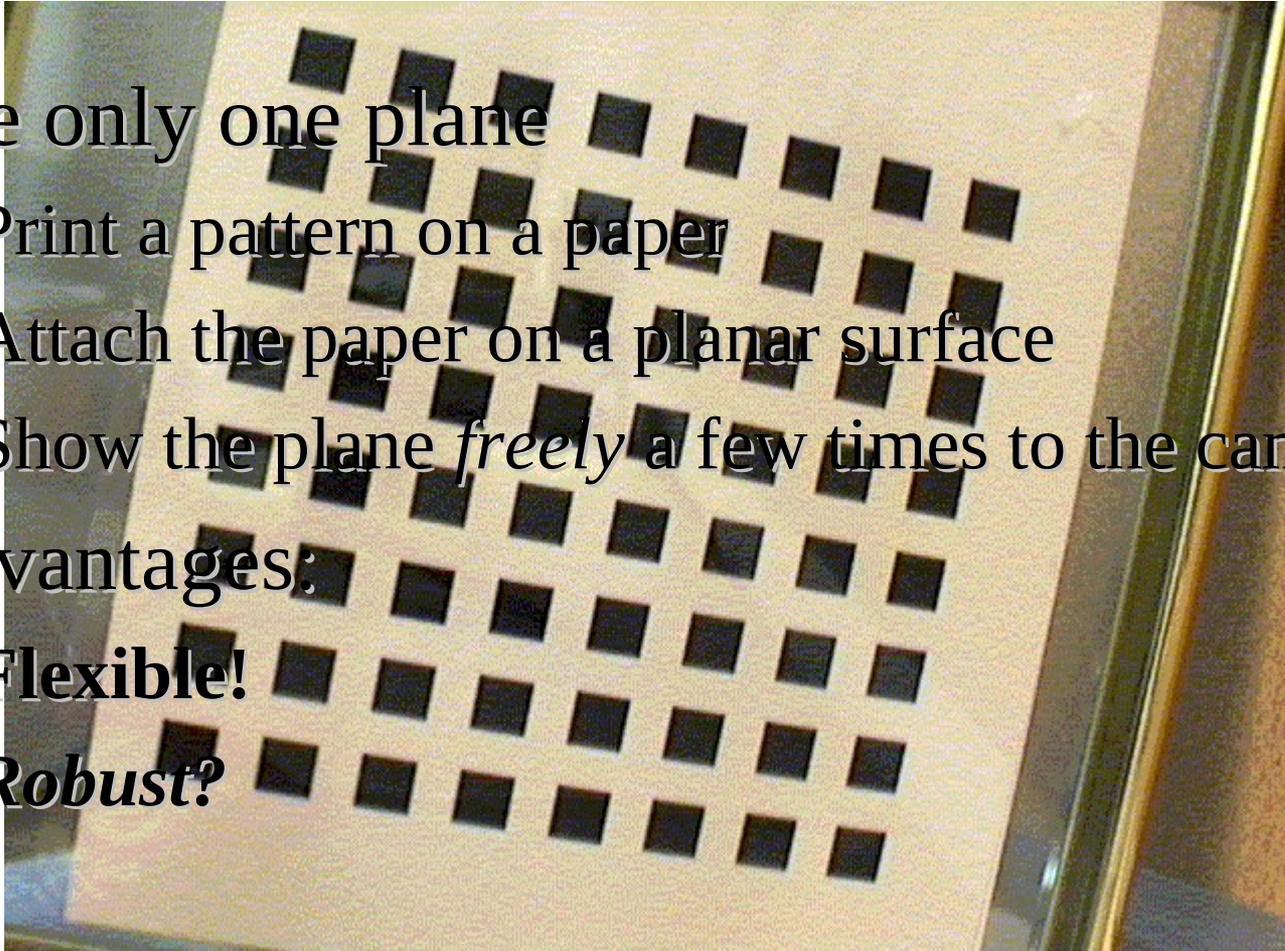


# Self-calibration



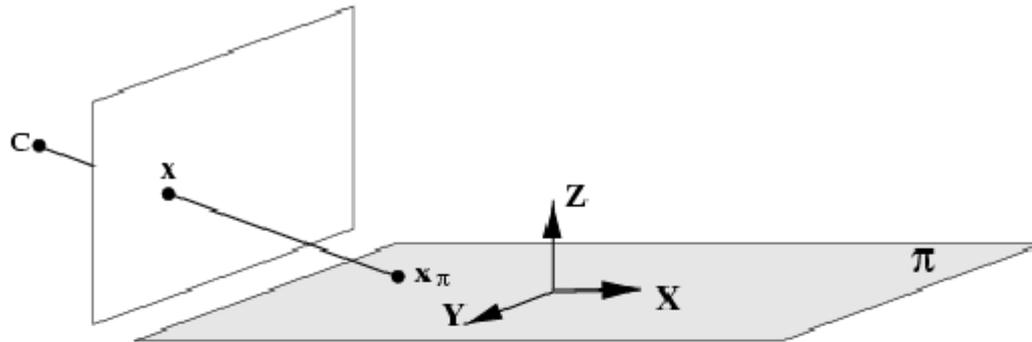
# Método de Zhang

- Use only one plane
  - Print a pattern on a paper
  - Attach the paper on a planar surface
  - Show the plane *freely* a few times to the camera
- Advantages:
  - **Flexible!**
  - ***Robust?***



# Plane projective transformations

---



Choose the world coordinate system such that the plane of the points has zero z coordinate.

Then the  $3 \times 4$  matrix  $P$  reduces to

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

which is a  $3 \times 3$  matrix representing a general plane to plane projective transformation.

## Example 1: Removing Perspective Distortion

---

**Given:** the coordinates of four points on the scene plane

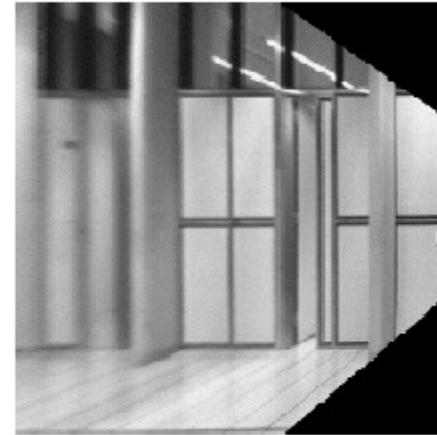
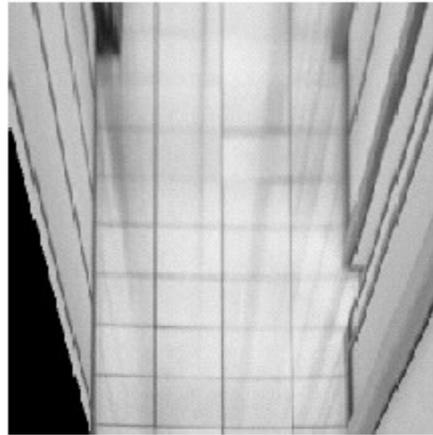
**Find:** a projective rectification of the plane



- This **rectification** does not require knowledge of **any** of the camera's parameters or the pose of the plane.
- It is not always necessary to know coordinates for four points.

## Example 2: Synthetic Rotations

---



**original**

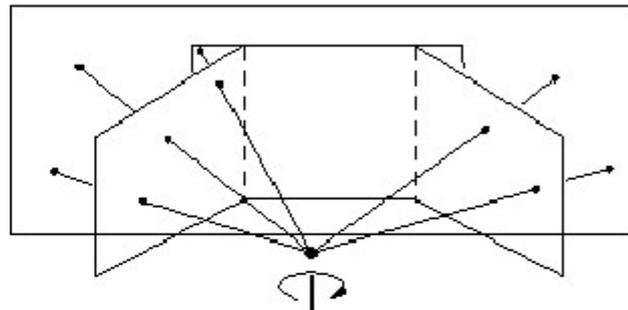
The synthetic images are produced by projectively warping the original image so that four corners of an imaged rectangle map to the corners of a rectangle. Both warpings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

For a rotation  $H = KRK^{-1}$

# Choice of mosaic frame

---

Choose central image as reference



This produces the classic "bow-tie" mosaic.



# Proyección de un plano

Por conveniencia, se asume el plano en  $z=0$

$$\begin{aligned} s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \\ &= \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} . \end{aligned}$$

- La relación entre puntos detectados en la imagen y puntos del modelo está dada por:

$$s\tilde{\mathbf{m}} = H\tilde{\mathbf{M}} \quad \text{con} \quad H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

# ¿Qué se obtiene de una imagen?

Se conoce  $H = [h_1 \quad h_2 \quad h_3]$

Como  $\mathbf{H}$  definida a menos de un factor de escala:

$$\lambda [h_1 \quad h_2 \quad h_3] = \mathbf{A} [r_1 \quad r_2 \quad t]$$

Por lo tanto:

$$\begin{aligned} h_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} h_2 &= 0 \\ h_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} h_1 &= h_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} h_2 \end{aligned}$$

# Sistema lineal de ecuaciones

- Se define  $B = A^{-T} A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$  ← simétrica

- Definiendo  $b = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]$

- Se reescribe

$$\begin{aligned} h_1^T A^{-T} A^{-1} h_2 &= 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2 \end{aligned}$$

como un sistema lineal

$$\mathbf{M}\mathbf{b} = 0$$

# Extracción de parámetros

Intrínsecos:

$$v_0 = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2)$$
$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11}$$
$$\alpha = \sqrt{\lambda / B_{11}}$$
$$\beta = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)}$$
$$\gamma = -B_{12}\alpha^2\beta / \lambda$$
$$u_0 = \gamma v_0 / \beta - B_{13}\alpha^2 / \lambda .$$

Extrínsecos:

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \mathbf{h}_1$$
$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1} \mathbf{h}_2$$
$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$
$$\mathbf{t} = \lambda \mathbf{A}^{-1} \mathbf{h}_3$$

# Solución cerrada

- Se muestra el plano en  $n$  orientaciones diferentes ( $n > 1$ )
- Se estiman las  $n$  homografías
- Se encuentran los parámetros intermedios
- Se extraen los parámetros intrínsecos
- Se calculan los extrínsecos para cada damero

# Refinamiento de la solución

- Dadas  $n$  imágenes del damero y  $m$  puntos por damero se minimiza

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)\|^2$$

¿Cuántas imágenes se precisan para  
calibrar?

# Distorsión

La imagen puede verse distorsionada.  
Muy común en lentes de bajo largo focal



barrel



pin-cushion

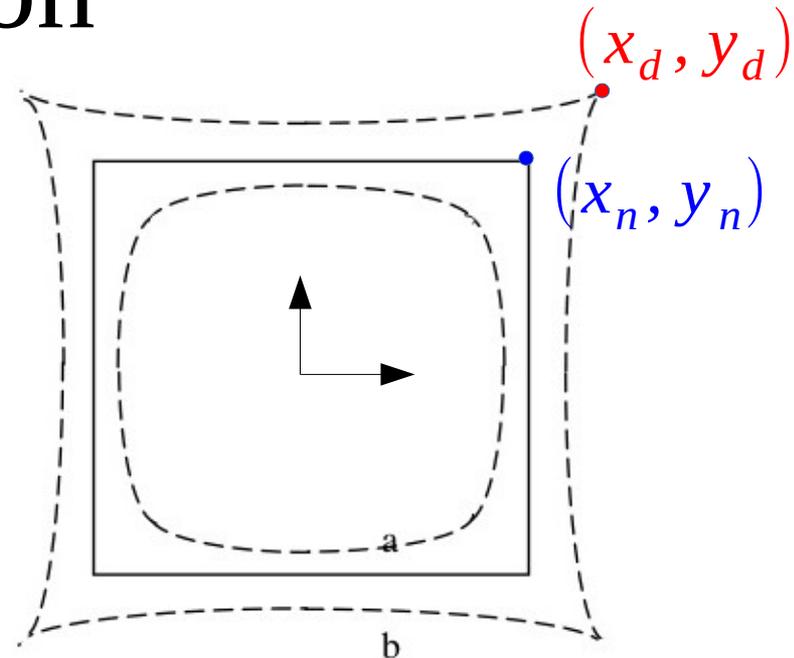


fisheye

# Proyección considerando distorsión

## 1- Proyección en perspectiva

$$X_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \longrightarrow x_n = \frac{x_c}{z_c} \quad y_n = \frac{y_c}{z_c}$$



## 2- Corrección distorsión

$$x_d = (1 + L(r)) x_n$$

$$y_d = (1 + L(r)) y_n$$

$$L(r) = k_1 r^2 + k_2 r^4$$

## 3- Conversión a píxeles y cambio de eje de coordenadas

$$K = \begin{bmatrix} \alpha_x & s & u_0 \\ & \alpha_y & v_0 \\ & & 1 \end{bmatrix}$$

$$x_p = \alpha_x x_d + s y_d + u_0$$

$$y_p = \alpha_y y_d + v_0$$

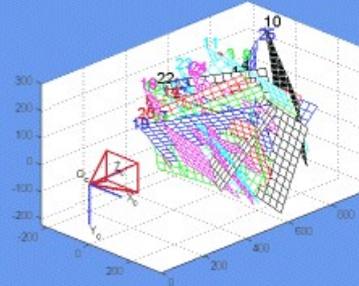
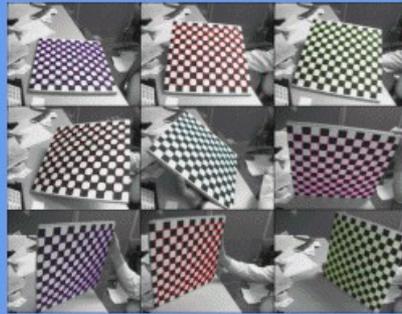
# Refinamiento de la solución

- En caso de haber distorsión se agregan los parámetros del modelo de distorsión en la minimización

$$\sum_{i=1}^n \sum_{j=1}^m \| \mathbf{m}_{ij} - \check{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j) \|^2$$

# Implementaciones

## *Camera Calibration Toolbox for Matlab*

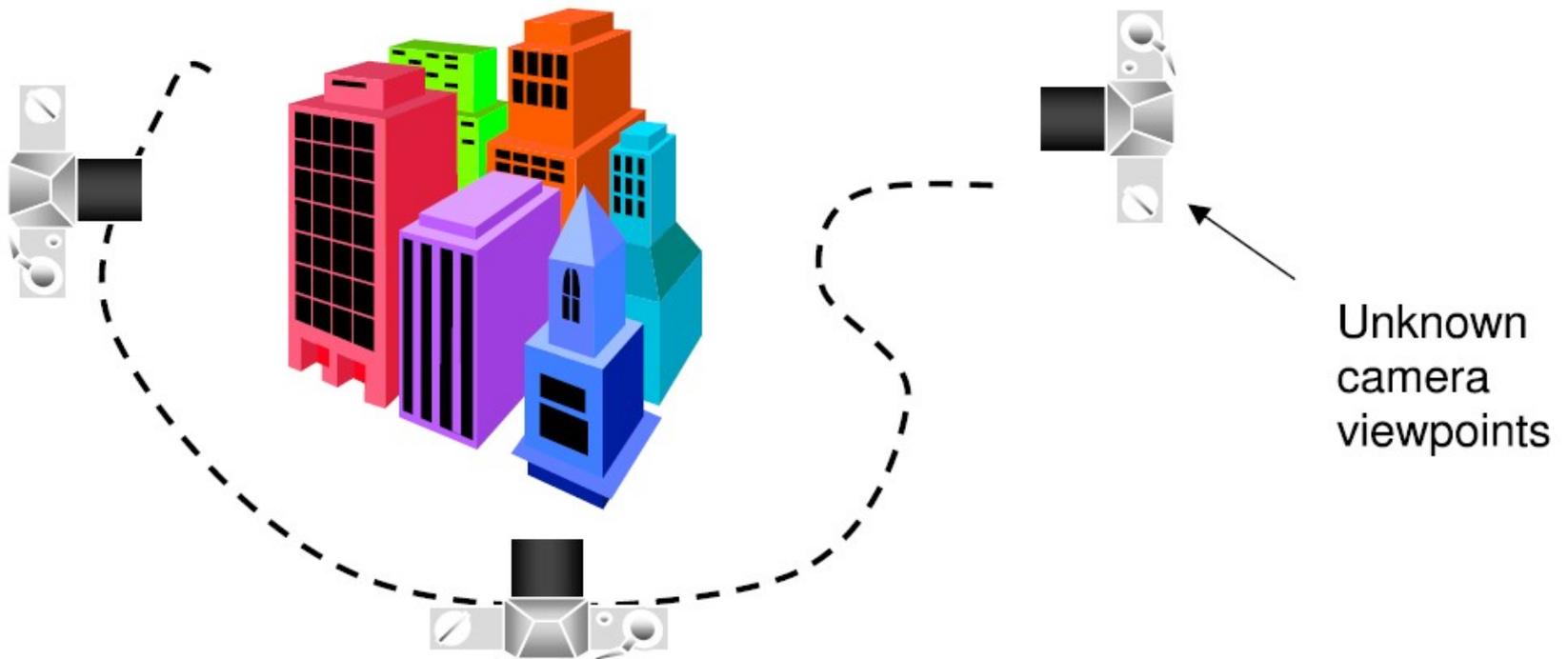


Ejemplo de OpenCV

# Geometría de dos cámaras

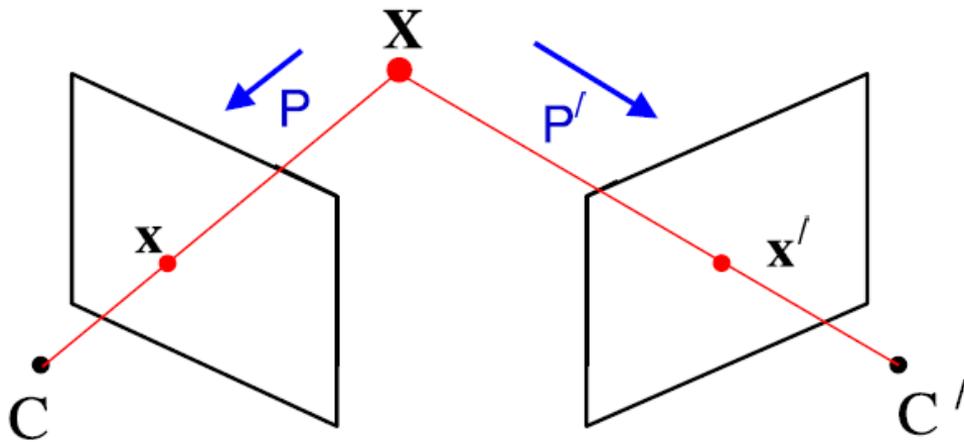
- Interesa responder
  - Dada la proyección de un punto en una cámara, ¿Dónde se ve en la otra?
  - ¿Cuál es la posición relativa de las cámaras?
  - ¿Cómo extraigo la información espacial de la escena?

# Reconstrucción de una escena



# Notación

$$\mathbf{x} = P\mathbf{X} \quad \mathbf{x}' = P'\mathbf{X}$$



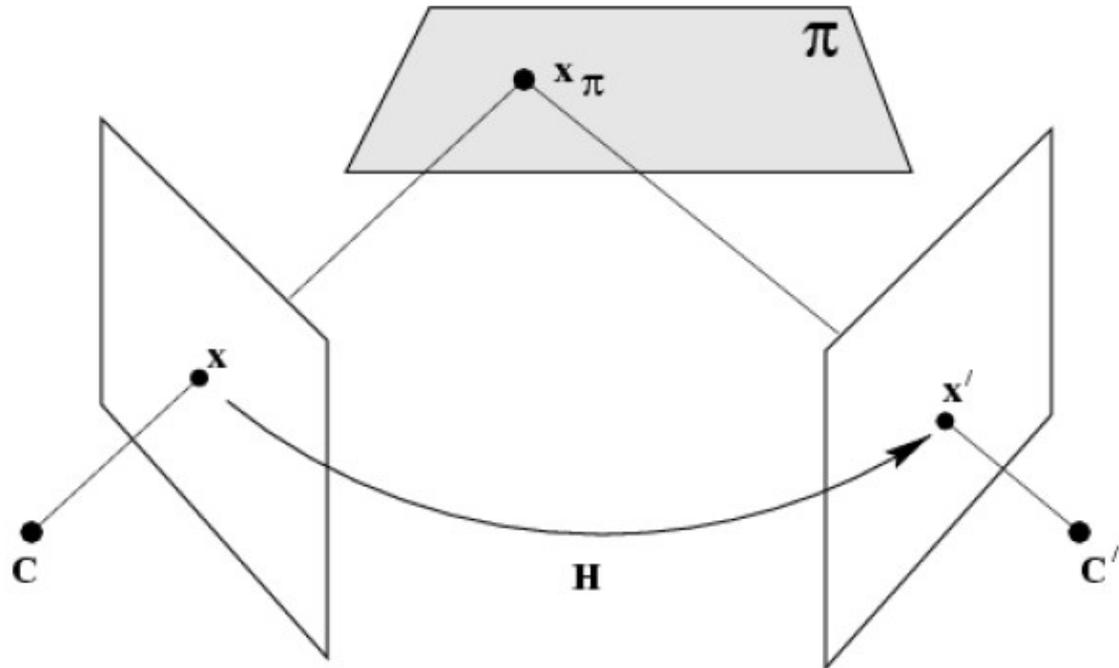
$P$  :  $3 \times 4$  matrix

$\mathbf{X}$  : 4-vector

$\mathbf{x}$  : 3-vector

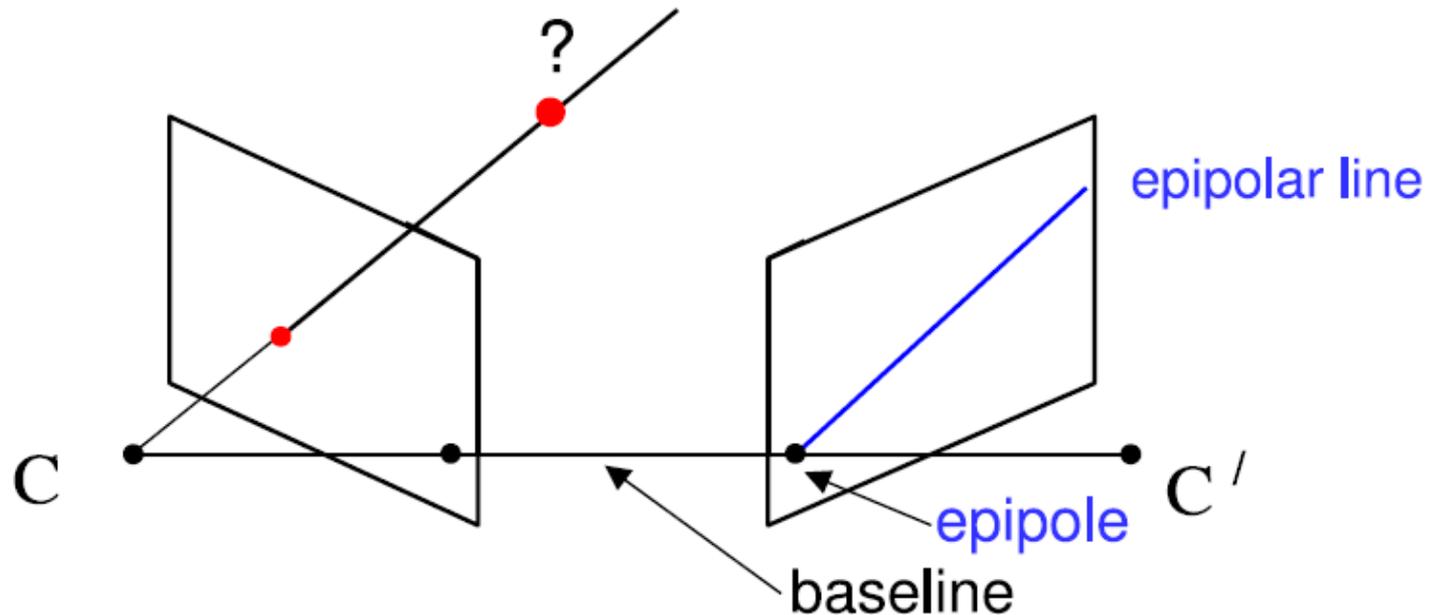
# Imágenes de un plano auxiliar

$$\begin{aligned} \mathbf{x} &= H_{1\pi} \mathbf{x}_\pi \\ \mathbf{x}' &= H_{2\pi} \mathbf{x}_\pi \\ &= H_{2\pi} H_{1\pi}^{-1} \mathbf{x} \\ &= H \mathbf{x} \end{aligned}$$



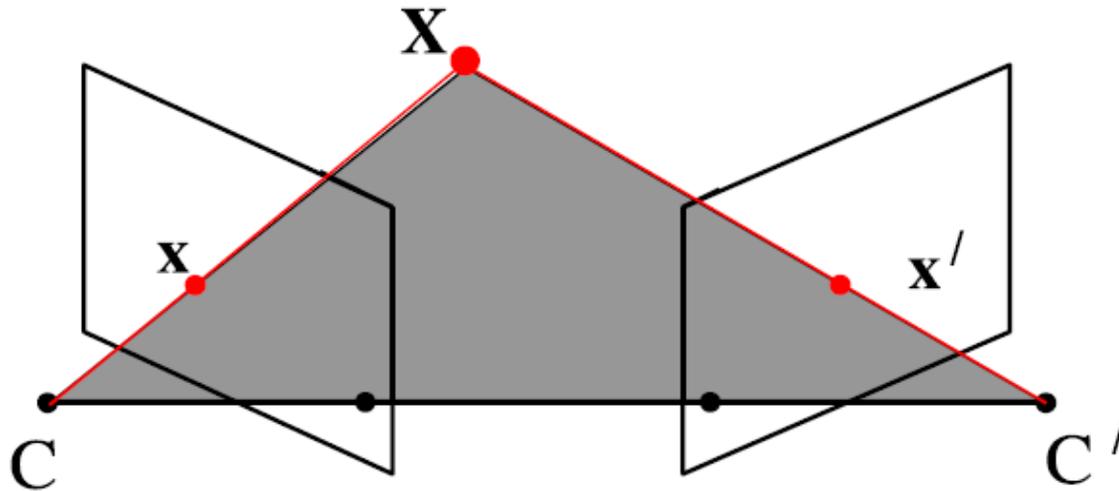
Existe una homografía que vincula las proyecciones en ambas imágenes. Es posible estimarla a partir de la correspondencia de cuatro puntos.

# Geometría Epipolar



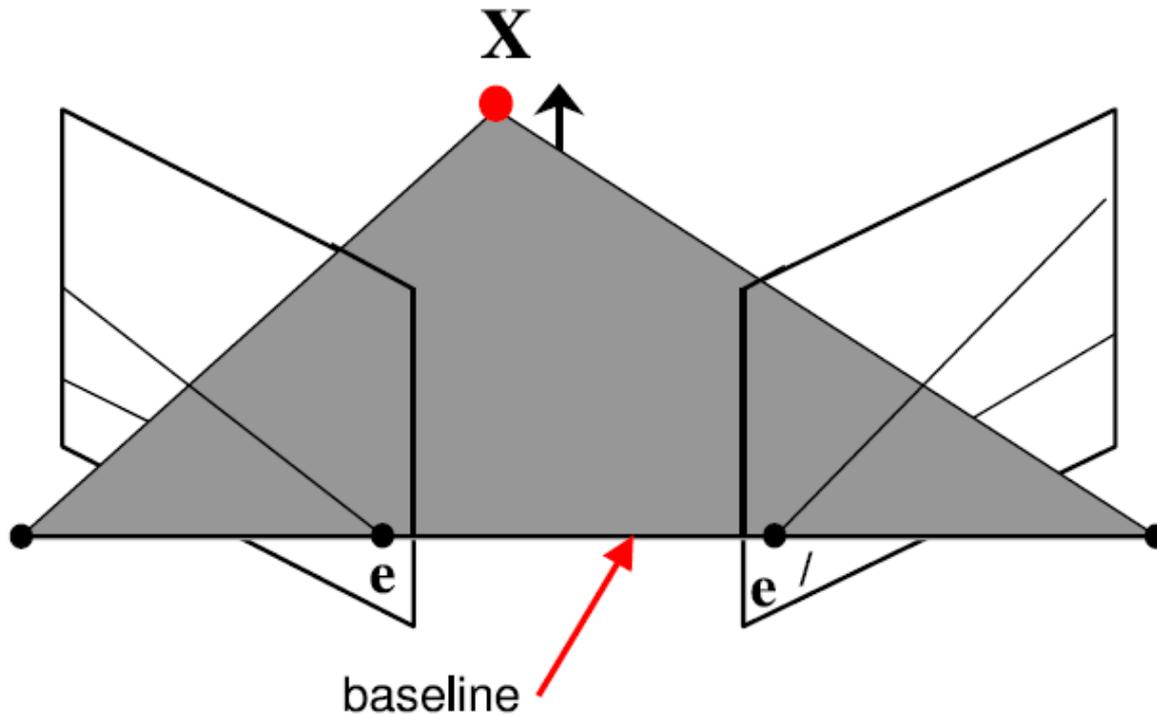
- Dado un punto en una imagen, éste determina una línea (línea epipolar) en la otra imagen.
- Su correspondiente forma parte de la línea epipolar

# Plano epipolar



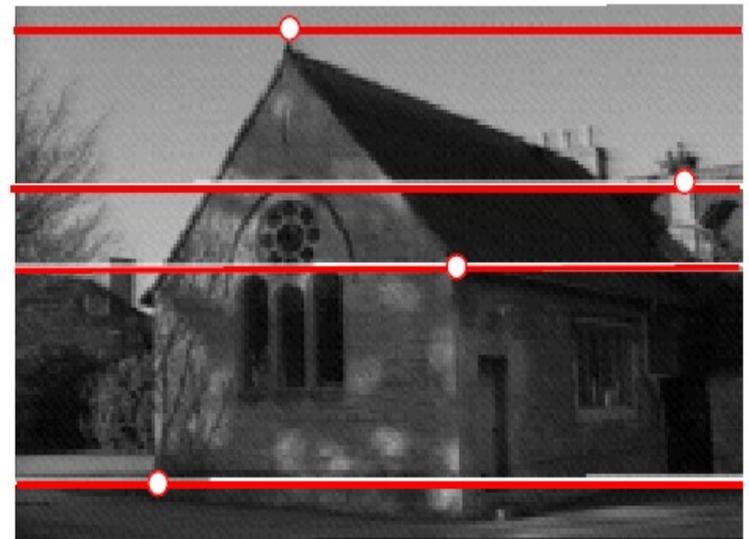
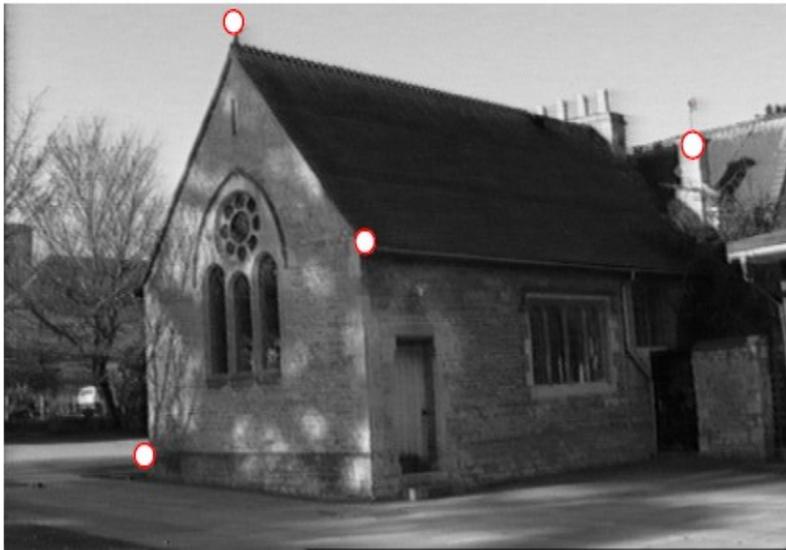
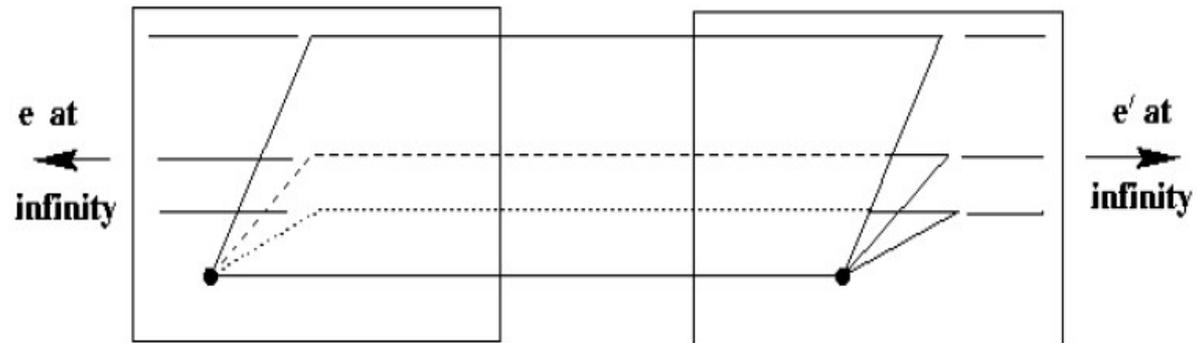
Los centros de las cámaras, el punto de la escena así como sus proyecciones en ambas cámaras, forman parte de un mismo plano.

# Lapiz epipolar (epipolar pencil)

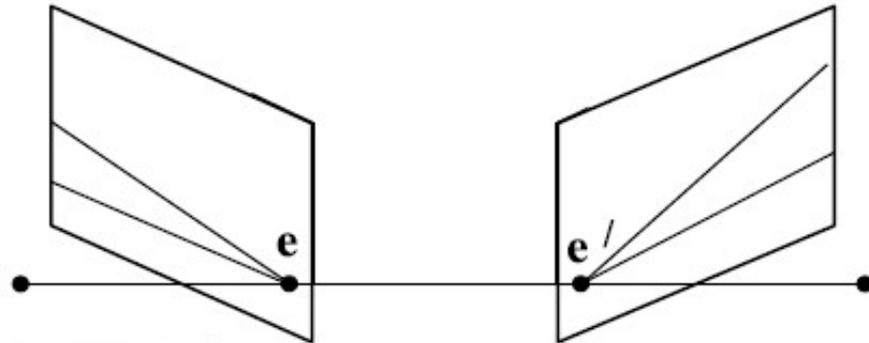


- Al variar la posición del punto  $X$ , los planos epipolares asociados rotan en torno al “baseline”.
- Todas las líneas epipolares se intersectan en el epipolo.

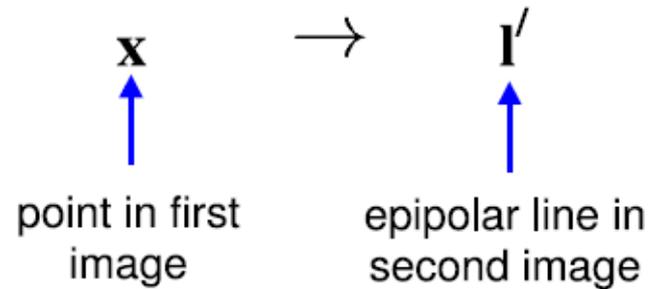
# Ejemplo: cámaras paralelas



# Ejemplo: cámaras convergentes



# Álgebra de la geometría epipolar



- Informalmente se vio que el mapeo no depende de la estructura de la escena.
- El mapeo es lineal y está dado por:  $\mathbf{l}' = \mathbf{F}\mathbf{x}$   
F es una matriz de 3x3, la **matriz fundamental**
- Puntos correspondientes cumplen:  $\mathbf{x}'^\top \mathbf{F}\mathbf{x} = 0$

# Propiedad 1

## Homogeneous notation for lines

---

Recall that a point  $(x, y)$  in 2D is represented by the homogeneous 3-vector  $\mathbf{x} = (x_1, x_2, x_3)^\top$ , where  $x = x_1/x_3, y = x_2/x_3$

A **line** in 2D is represented by the homogeneous 3-vector

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$$

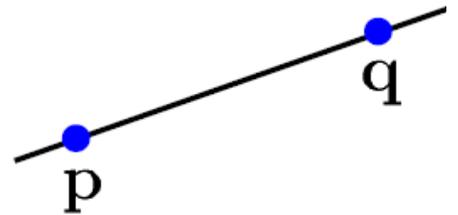
which is the line  $l_1x + l_2y + l_3 = 0$ .

# Propiedad 2

- The line  $\mathbf{l}$  through the two points  $\mathbf{p}$  and  $\mathbf{q}$  is  $\mathbf{l} = \mathbf{p} \times \mathbf{q}$

Proof

$$\mathbf{l} \cdot \mathbf{p} = (\mathbf{p} \times \mathbf{q}) \cdot \mathbf{p} = 0 \quad \mathbf{l} \cdot \mathbf{q} = (\mathbf{p} \times \mathbf{q}) \cdot \mathbf{q} = 0$$



# Propiedad 3

The vector product  $\mathbf{v} \times \mathbf{x}$  can be represented as a matrix multiplication

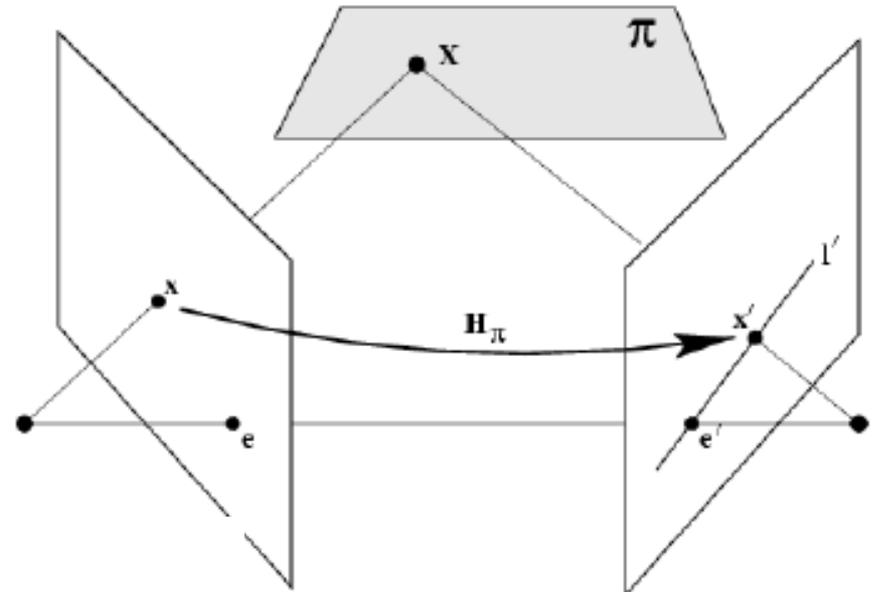
$$\mathbf{v} \times \mathbf{x} = \begin{pmatrix} v_2x_3 - v_3x_2 \\ v_3x_1 - v_1x_3 \\ v_1x_2 - v_2x_1 \end{pmatrix} = [\mathbf{v}]_{\times} \mathbf{x}$$

where

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

- $[\mathbf{v}]_{\times}$  is a  $3 \times 3$  skew-symmetric matrix of rank 2.
- $\mathbf{v}$  is the null-vector of  $[\mathbf{v}]_{\times}$ , since  $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_{\times} \mathbf{v} = \mathbf{0}$ .

# Matriz fundamental



Step 1: Point transfer via a plane  $x' = H_{\pi}x$

Step 2 : Construct the epipolar line  $l' = e' \times x' = [e']_{\times}x'$

$$l' = [e']_{\times}H_{\pi}x = Fx$$

$$F = [e']_{\times}H_{\pi}$$

This shows that  $F$  is a  $3 \times 3$  rank 2 matrix.

# Cálculo de la matriz fundamental

## Problem statement

---

Given:  $n$  corresponding points  $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i, i = 1, \dots, n\}$   
compute the fundamental matrix  $F$  such that

$$\mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i = 0 \quad 1 \leq i \leq n$$

## Solution

Each point correspondence  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  generates one constraint on  $F$

$$(x'_i \ y'_i \ 1) \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = 0$$

which may be written

$$x'x f_1 + x'y f_2 + x' f_3 + y'x f_4 + y'y f_5 + y' f_6 + x f_7 + y f_8 + f_9 = 0$$

# Cálculo de la matriz fundamental

Given Image pair



Find The fundamental matrix  $F$  and correspondences  $x_i \leftrightarrow x'_i$ .

- Compute image points
- Compute correspondences
- Compute epipolar geometry

# Cálculo de la matriz fundamental

Step 1: interest points

---



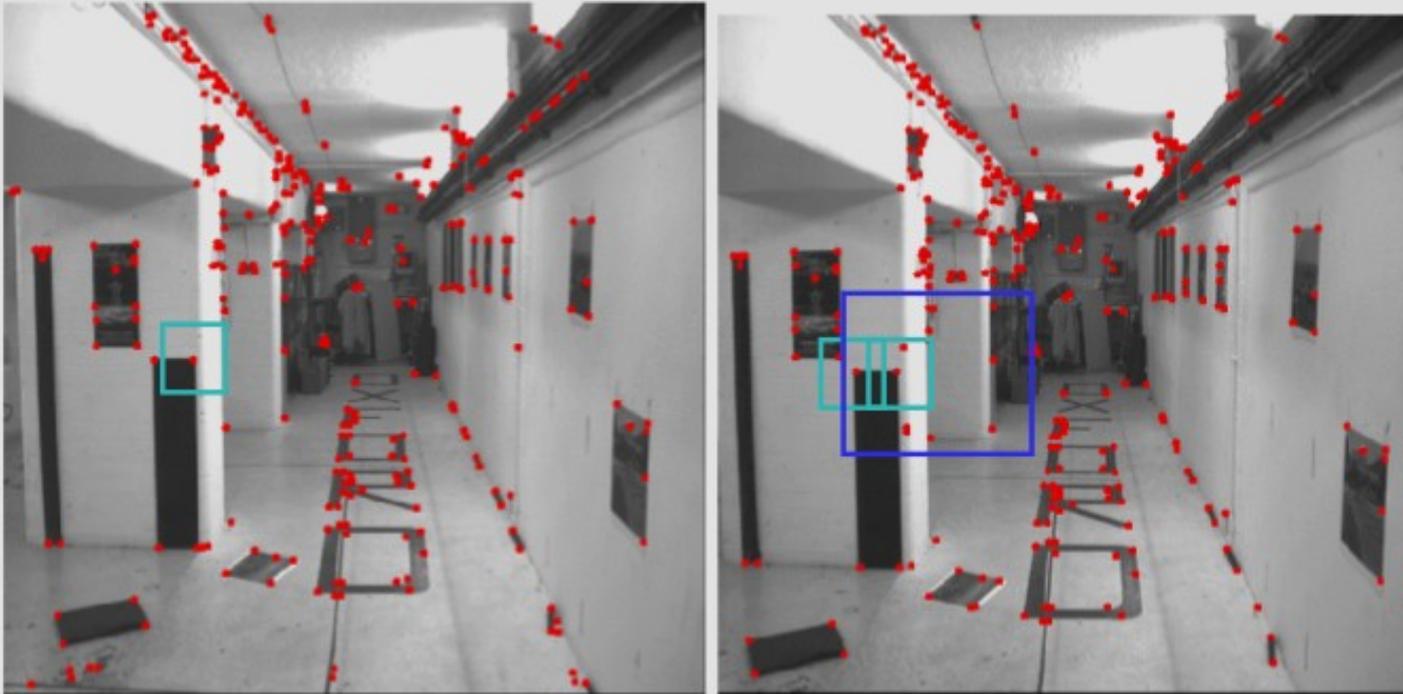
Harris corner detector

100's of points per image

# Cálculo de la matriz fundamental

Step 2b: match points – cross-correlate

---

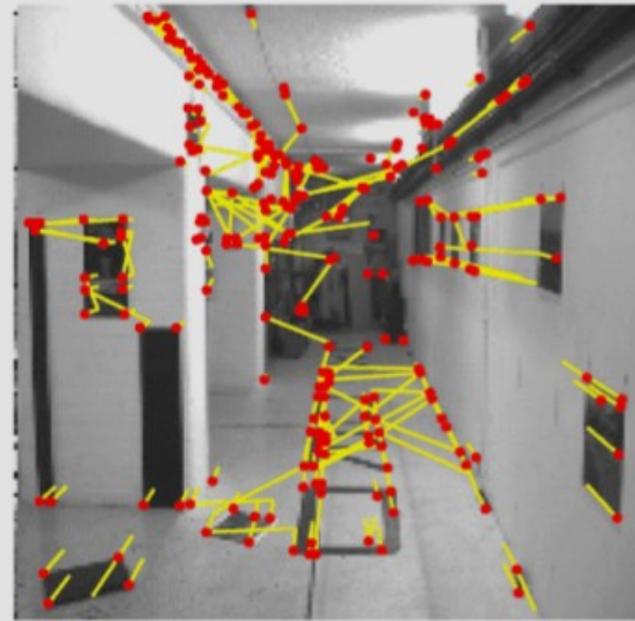


- cross-correlate on intensity neighbourhoods

# Cálculo de la matriz fundamental

## Correlation matching results

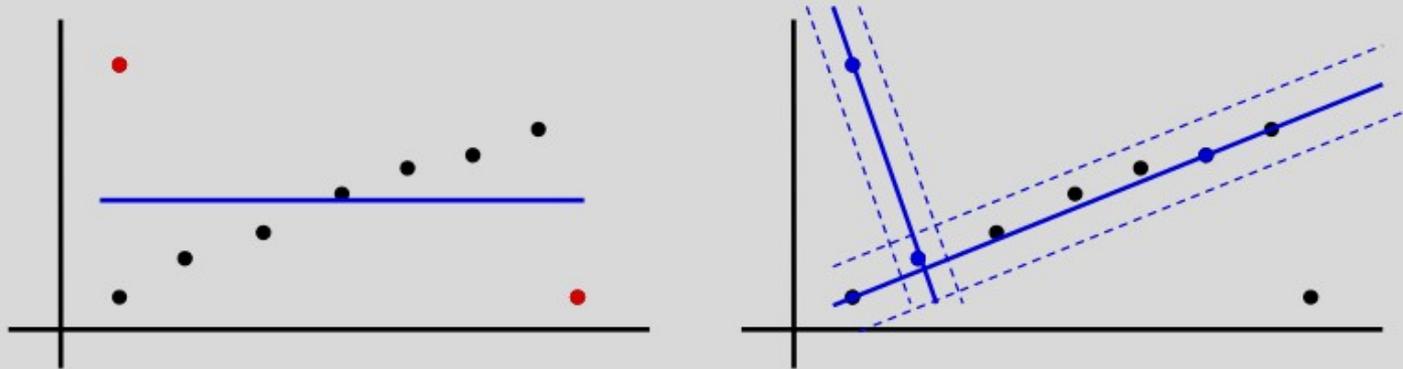
---



- Many wrong matches (10-50%), but enough to compute  $F$

# Estimación robusta - RANSAC

Fit a line to 2D data containing outliers



Repeat

1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

Choose the line with the largest number of inliers

- Compute least squares fit of line to inliers (regression)

# Estimación robusta matriz fundamental - RANSAC

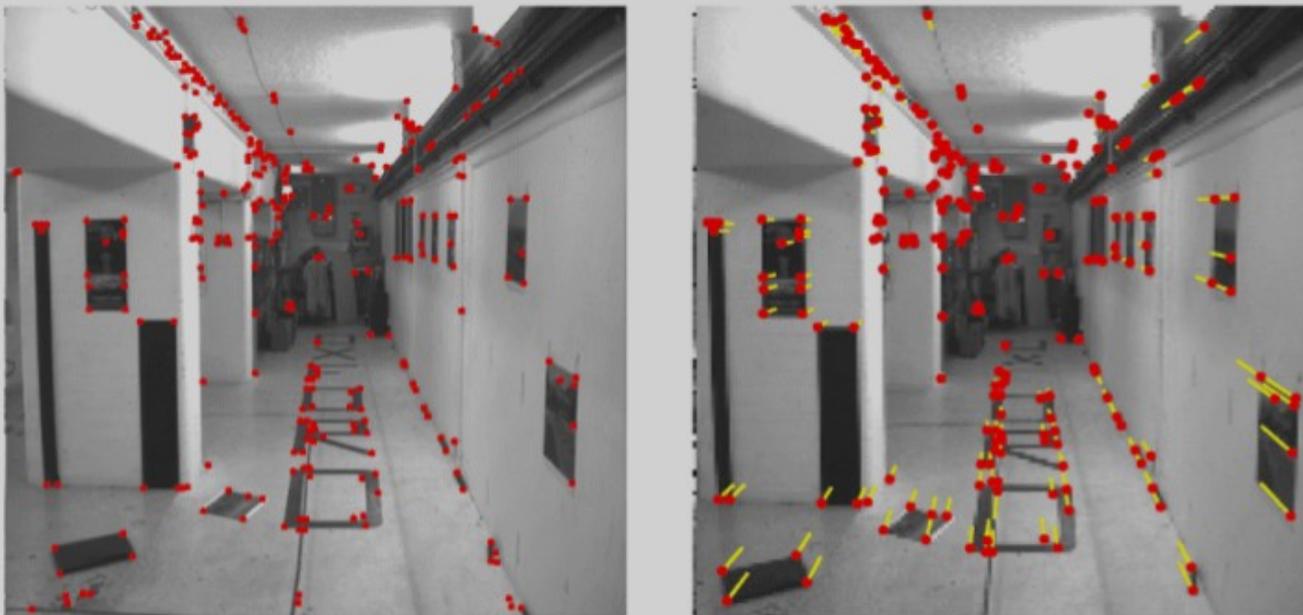
Repetir:

- 1- Seleccionar aleatoriamente 7 correspondencias
- 2- Calcular F
- 3- Determinar la cantidad de inliers:

$$d(x',l') < \text{margen} \quad \text{con } l' = Fx$$

Elegir F con la mayor cantidad de inliers

# Correspondencias consistentes con geometría epipolar



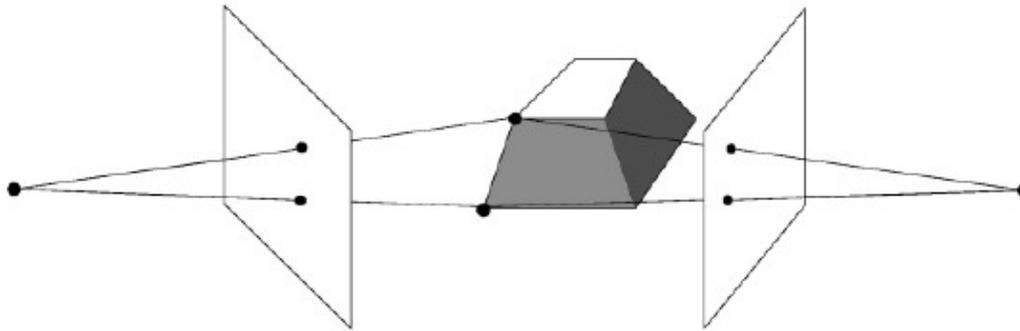
- Use **RANSAC** robust estimation algorithm
- Obtain correspondences  $x_i \leftrightarrow x'_i$  and  $F$

# Interludio musical

The Fundamental Matrix Song

The Ransac Song

# Reconstrucción a partir de dos vistas



# Ambigüedad en la reconstrucción

Given: image point correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ ,  
compute a reconstruction:

$$\{P, P', \mathbf{X}_i\} \quad \text{with} \quad \mathbf{x}_i = P\mathbf{X}_i \quad \mathbf{x}'_i = P'\mathbf{X}_i$$

Ambiguity

$$\mathbf{x}_i = P\mathbf{X}_i = P H(H)^{-1} \mathbf{X}_i = \tilde{P}\tilde{\mathbf{X}}_i$$

$$\mathbf{x}'_i = P'\mathbf{X}_i = P' H(H)^{-1} \mathbf{X}_i = \tilde{P}'\tilde{\mathbf{X}}_i$$

$\{\tilde{P}, \tilde{P}', \tilde{\mathbf{X}}_i\}$  is an equivalent **Projective Reconstruction**.

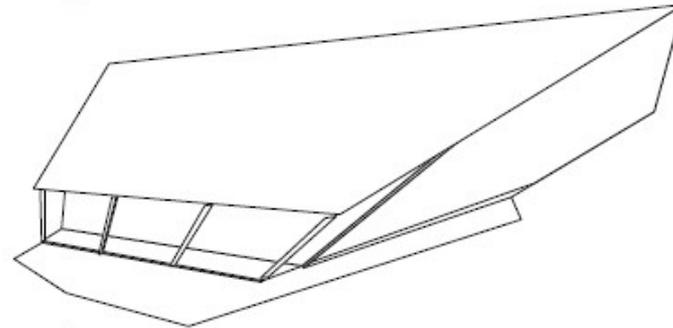
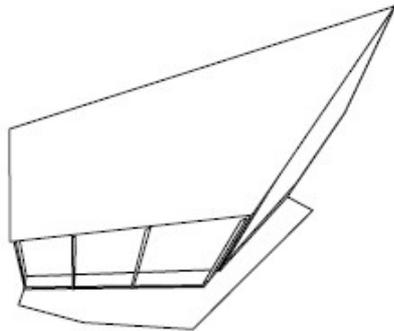
# Ambigüedad en la reconstrucción



# Reconstrucción proyectiva



a



b

Fig. 10.3. **Projective reconstruction.** (a) Original image pair. (b) 2 views of a 3D projective reconstruction of the scene. The reconstruction requires no information about the camera matrices, or information about the scene geometry. The fundamental matrix  $F$  is computed from point correspondences between the images, camera matrices are retrieved from  $F$ , and then 3D points are computed by triangulation from the correspondences. The lines of the wireframe link the computed 3D points.

# Reconstrucción afín

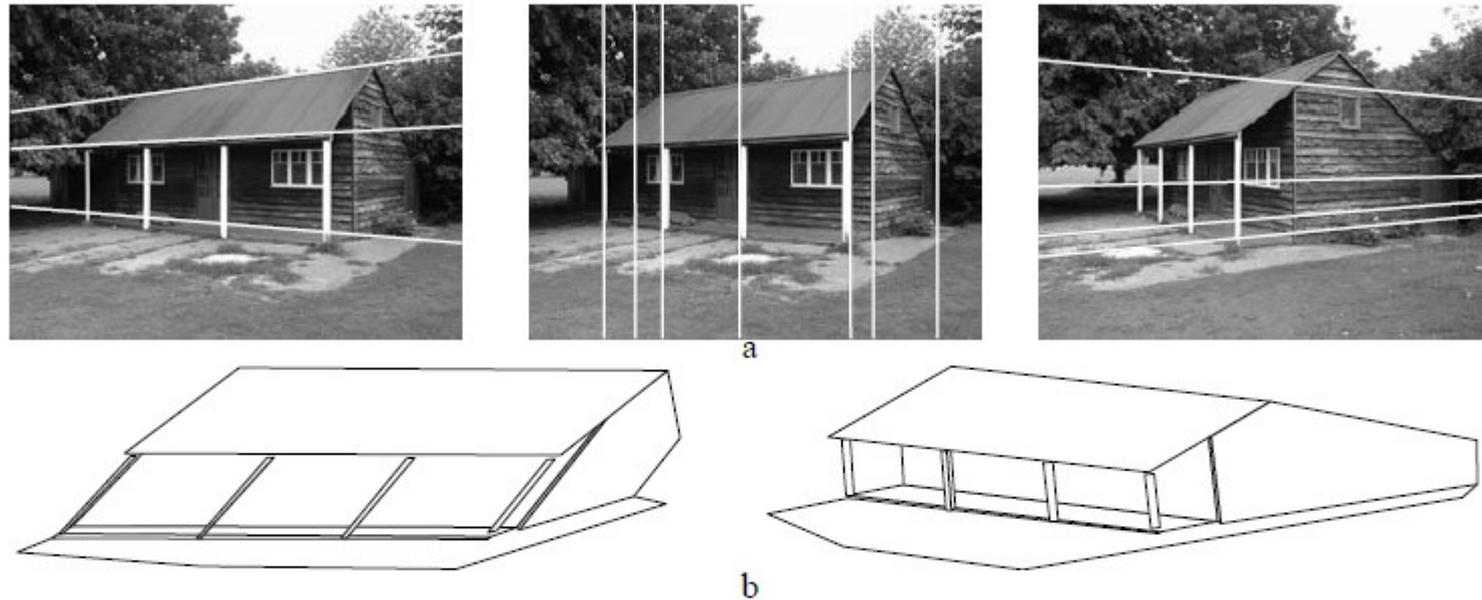
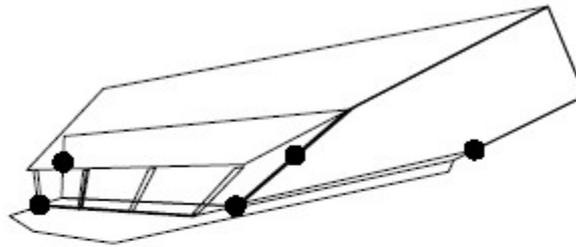


Fig. 10.4. **Affine reconstruction.** *The projective reconstruction of figure 10.3 may be upgraded to affine using parallel scene lines. (a) There are 3 sets of parallel lines in the scene, each set with a different direction. These 3 sets enable the position of the plane at infinity,  $\pi_\infty$ , to be computed in the projective reconstruction. The wireframe projective reconstruction of figure 10.3 is then affinely rectified using the homography (10.2). (b) Shows two orthographic views of the wireframe affine reconstruction. Note that parallel scene lines are parallel in the reconstruction, but lines that are perpendicular in the scene are not perpendicular in the reconstruction.*

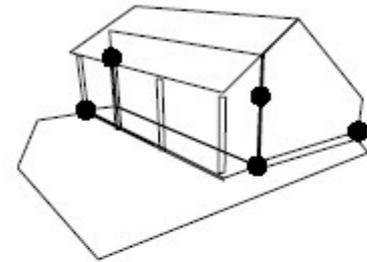
# Reconstrucción métrica



a



b



c

Fig. 10.6. **Direct reconstruction.** *The projective reconstruction of figure 10.3 may be upgraded to metric by specifying the position of five (or more) world points: (a) the five points used; (b) the corresponding points on the projective reconstruction of figure 10.3; (c) the reconstruction after the five points are mapped to their world positions.*

# Reconstrucción proyectiva en tres pasos

1. Compute the epipolar geometry (represented by the fundamental matrix  $F$ ) from point correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

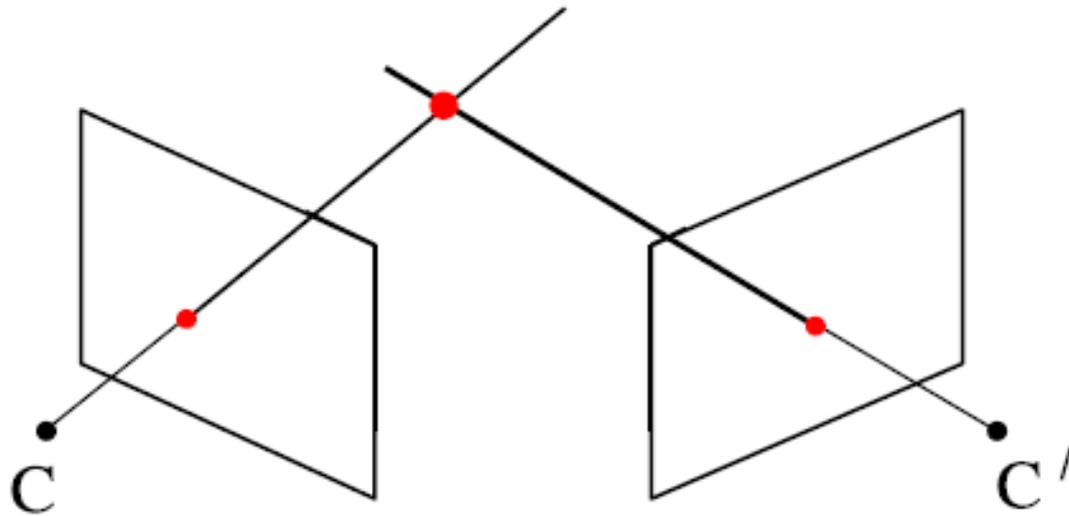
2. Compute the cameras (**motion**) from the matrix  $F$ . Obtain

$$P = [I \mid \mathbf{0}] \quad P' = [ [\mathbf{e}']_{\times} F \mid \mathbf{e}' ], \quad \text{where } F^{\top} \mathbf{e}' = \mathbf{0}$$

where  $P$  and  $P'$  are up to a projective ambiguity

3. Compute the 3D structure  $\mathbf{X}_i$  from the cameras  $P, P'$  and point correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  (**triangulation**)

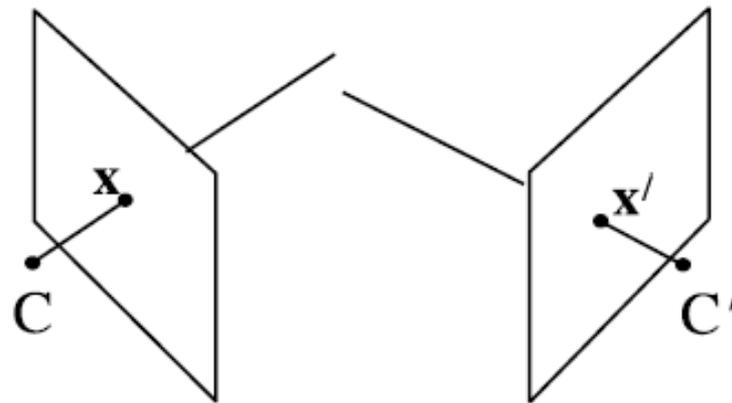
# Triangulación



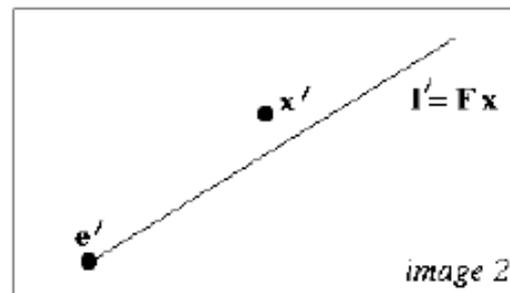
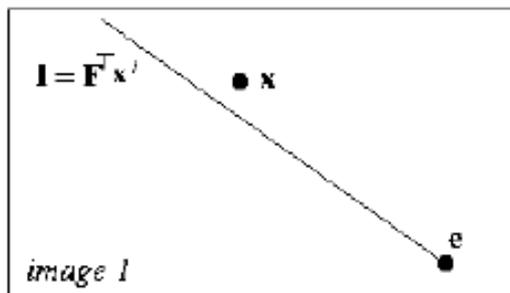
“Back projected points” se intersectan en el punto de la escena que origino las proyecciones

# Triangulación

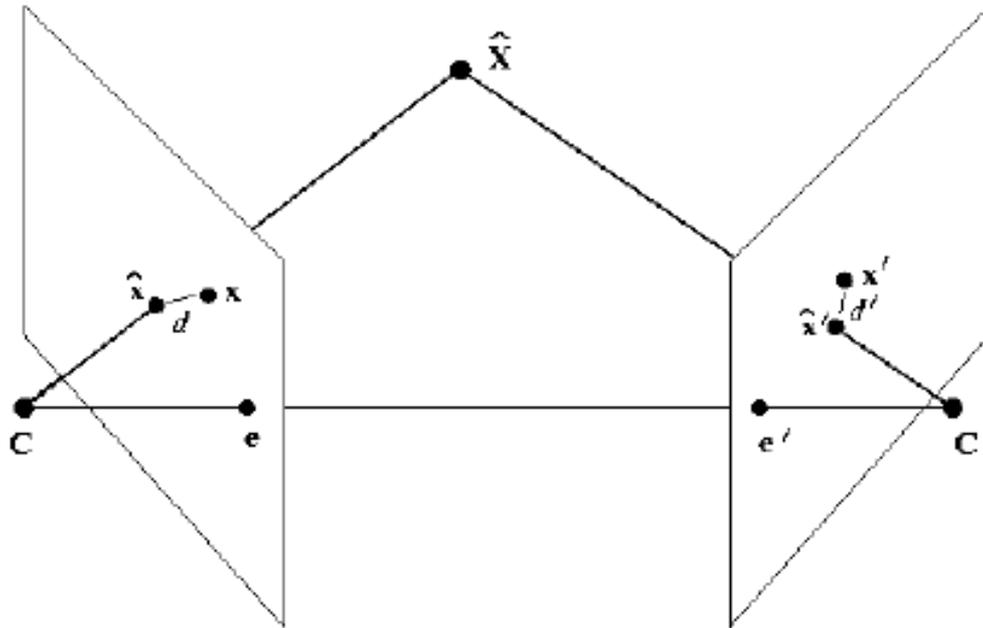
- Problema: el ruido hace que los rayos no se intersecten



rays are skew in space



# Solución estadística



$$\min_{\hat{X}} C(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

where  $d(*, *)$  is the Euclidean distance between the points.

# RECONSTRUCCIÓN ESTÉREO

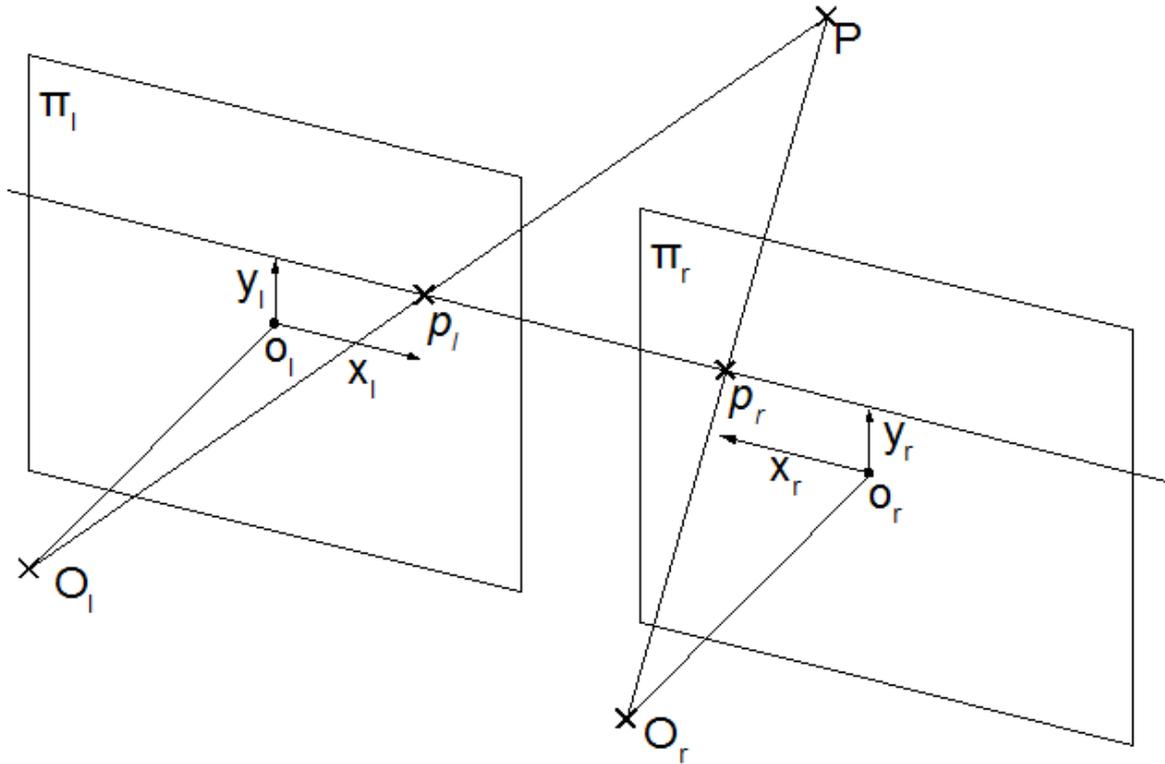
## Par estéreo

- Se quiere lograr una reconstrucción densa de la escena, para todos los puntos de la imagen
- No es trivial encontrar correspondencias para todos los puntos de la imagen izquierda en la derecha
- Para ayudar a los algoritmos que encuentran correspondencias se realiza una transformación de las imágenes para que la búsqueda sea eficiente

# RECONSTRUCCIÓN ESTÉREO

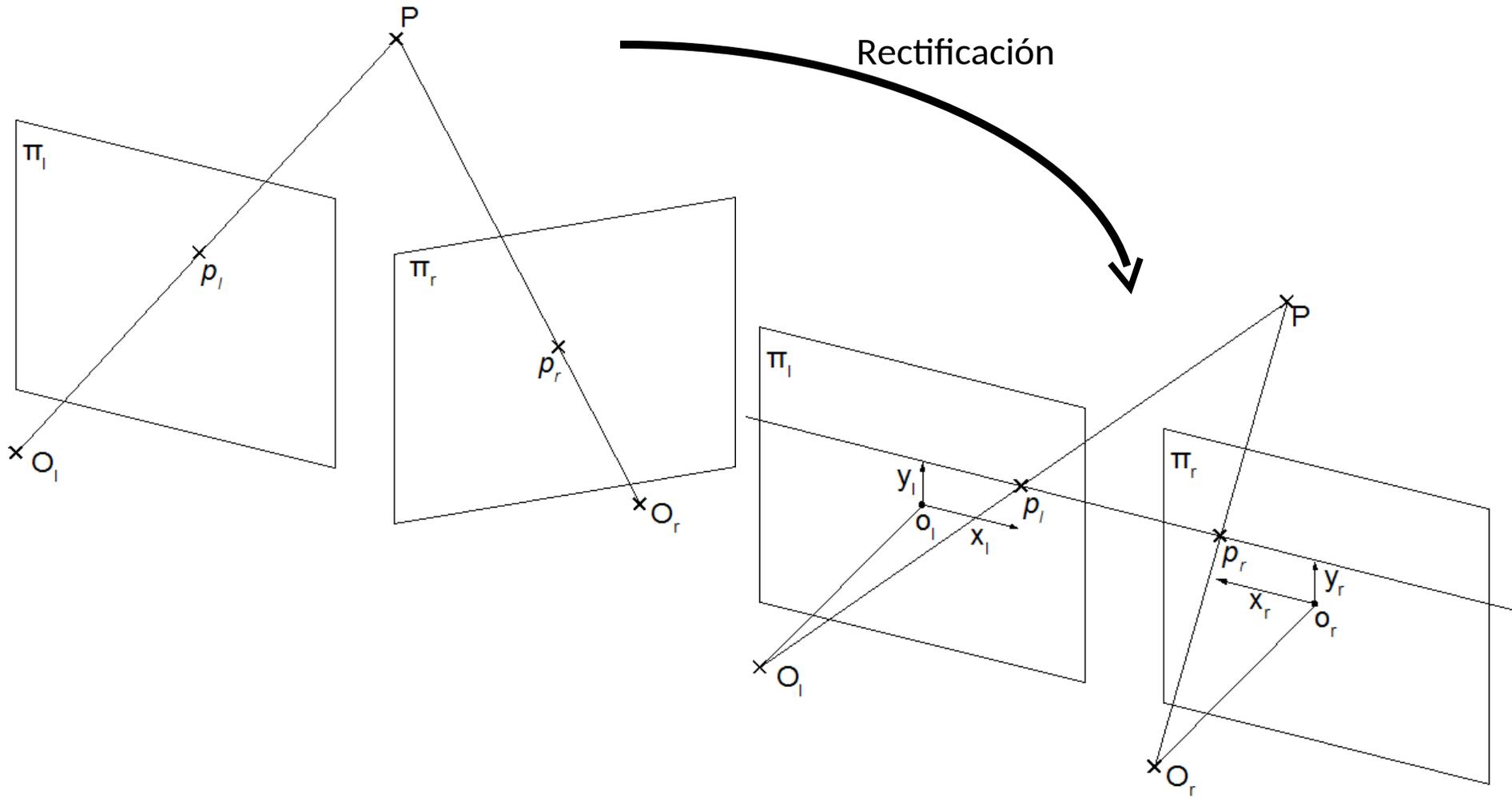
## Par estéreo

- Cámaras con sus planos de imagen coplanares y sus filas alineadas



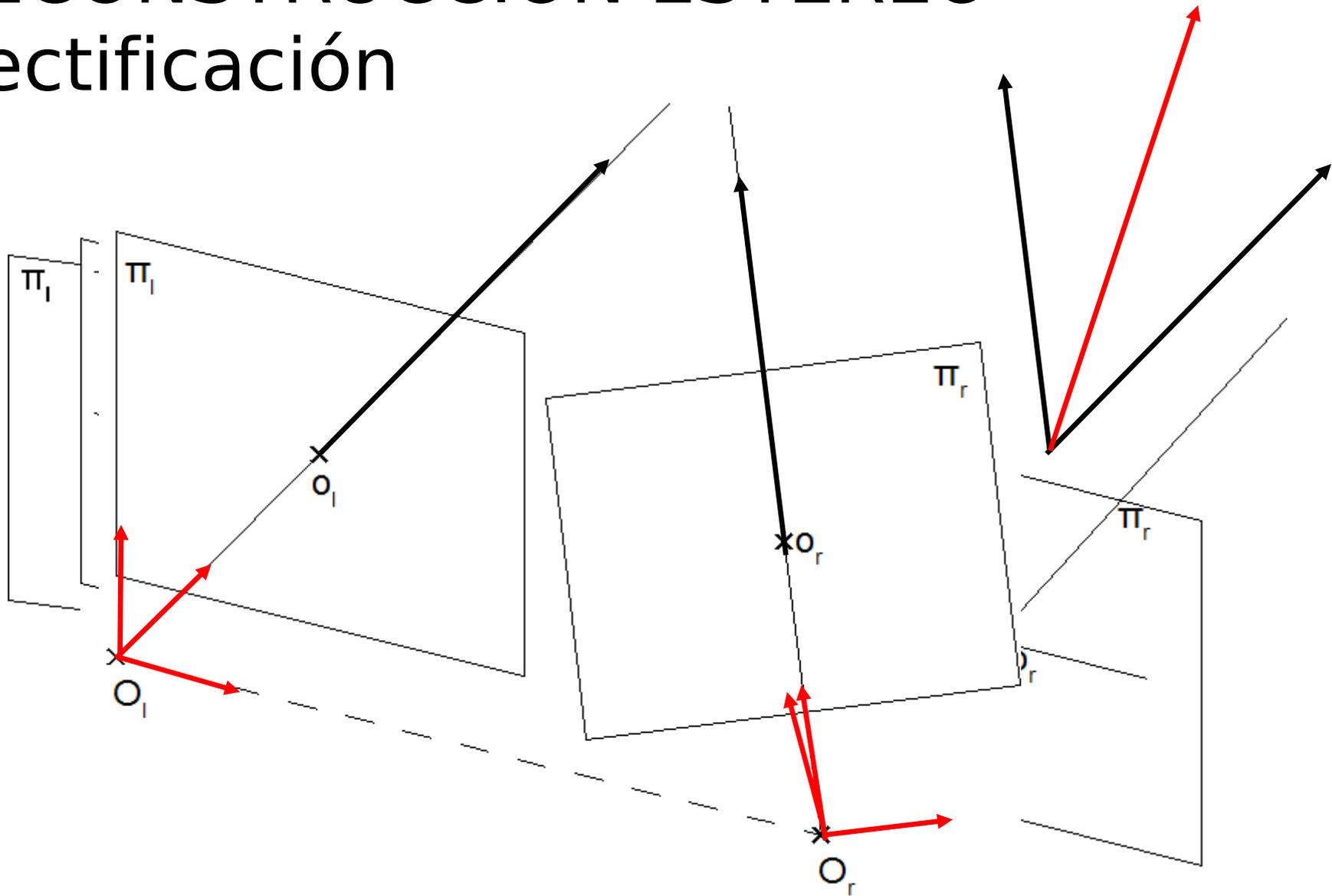
# RECONSTRUCCIÓN ESTÉREO

## Par estéreo real



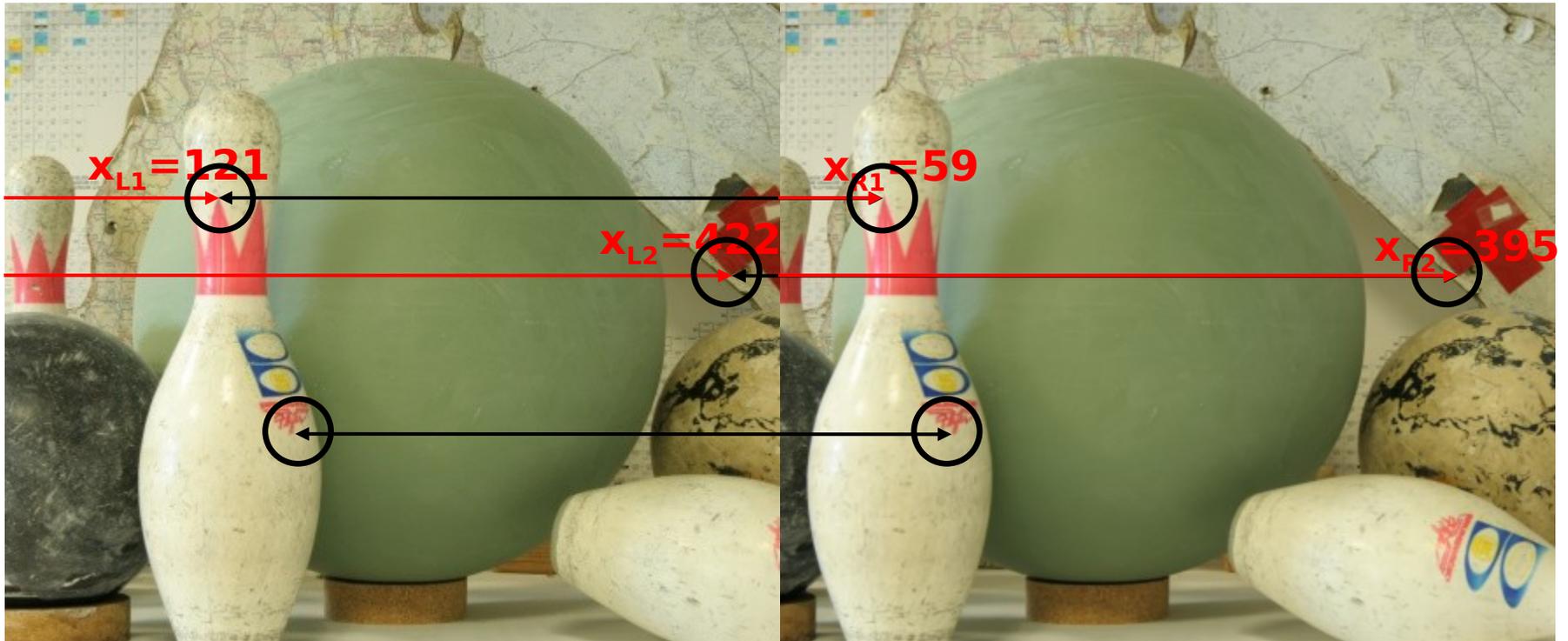
# RECONSTRUCCIÓN ESTÉREO

## Rectificación



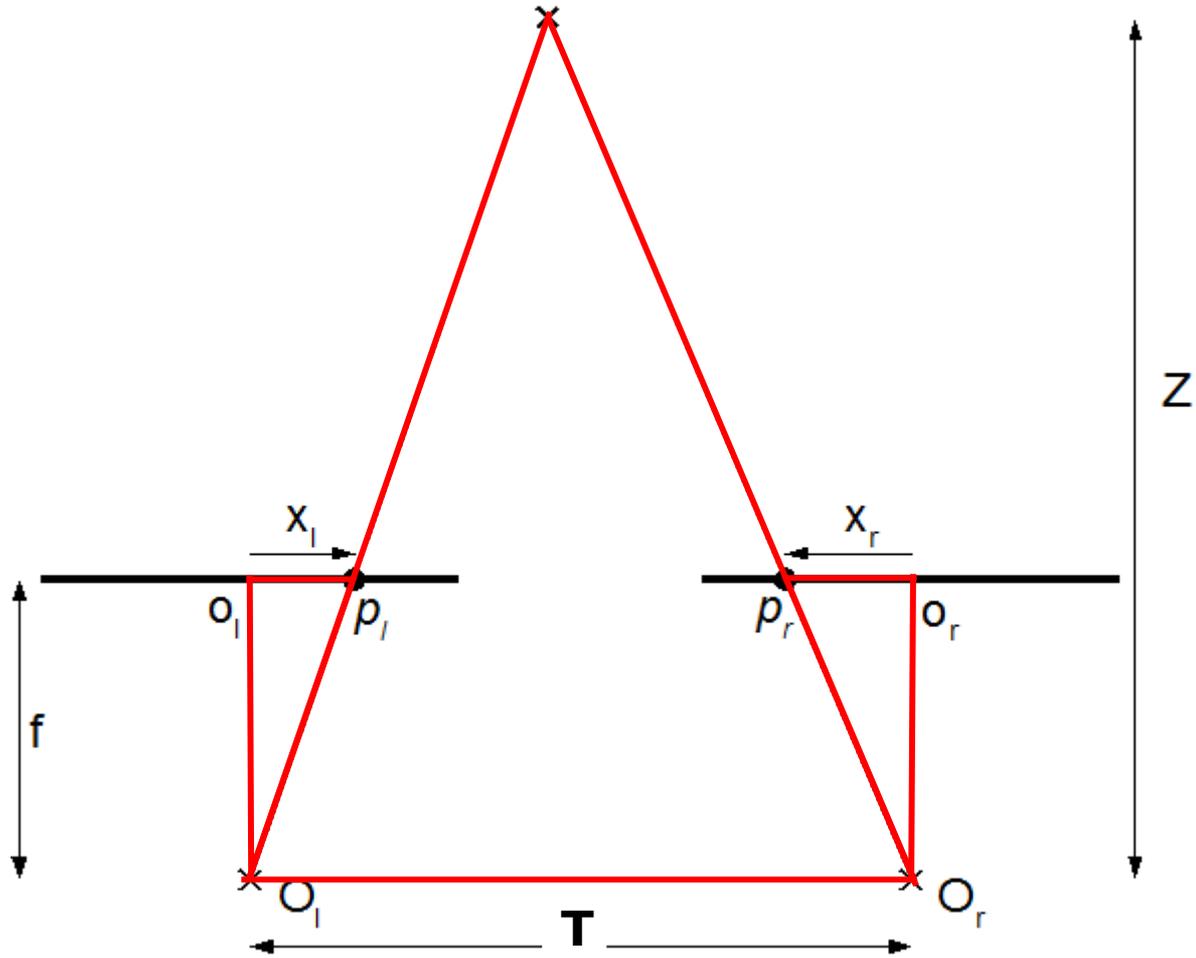
# RECONSTRUCCIÓN ESTÉREO

## Profundidad y disparidad



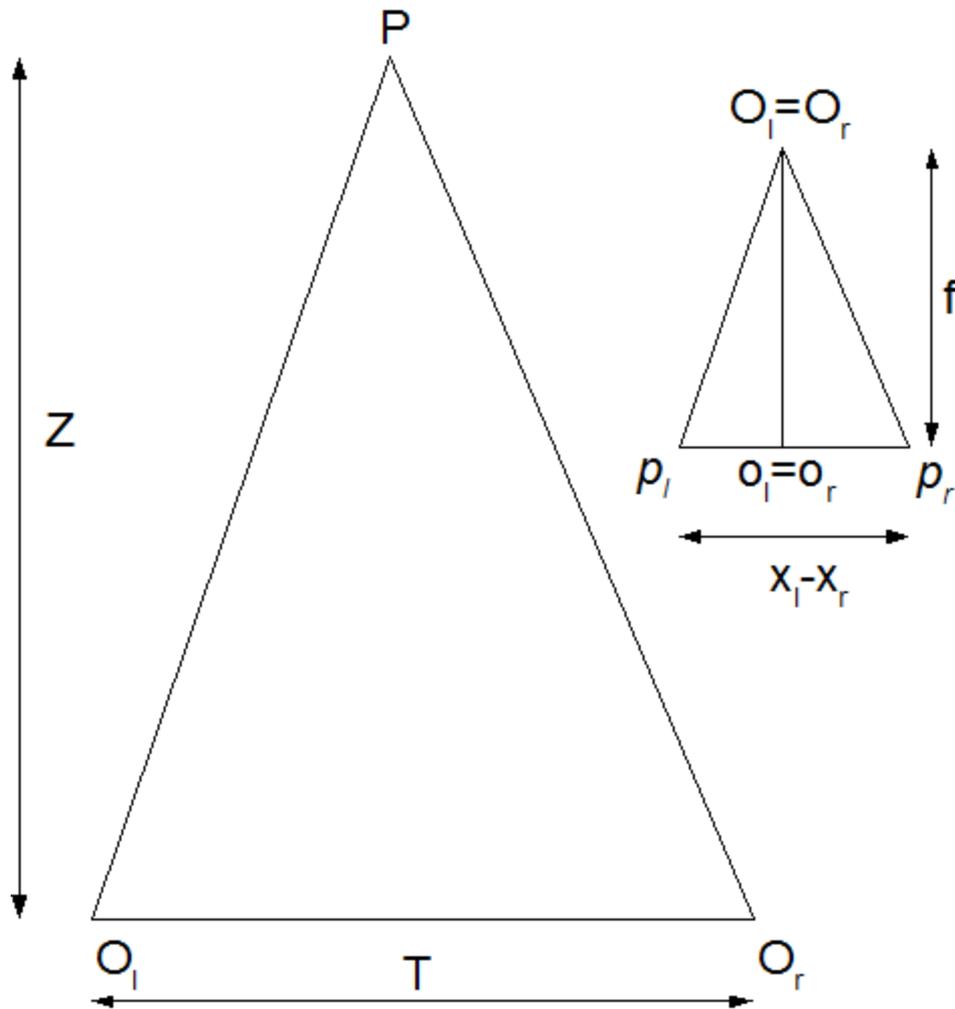
# RECONSTRUCCIÓN ESTÉREO

## Profundidad y disparidad



# RECONSTRUCCIÓN ESTÉREO

## Profundidad y disparidad

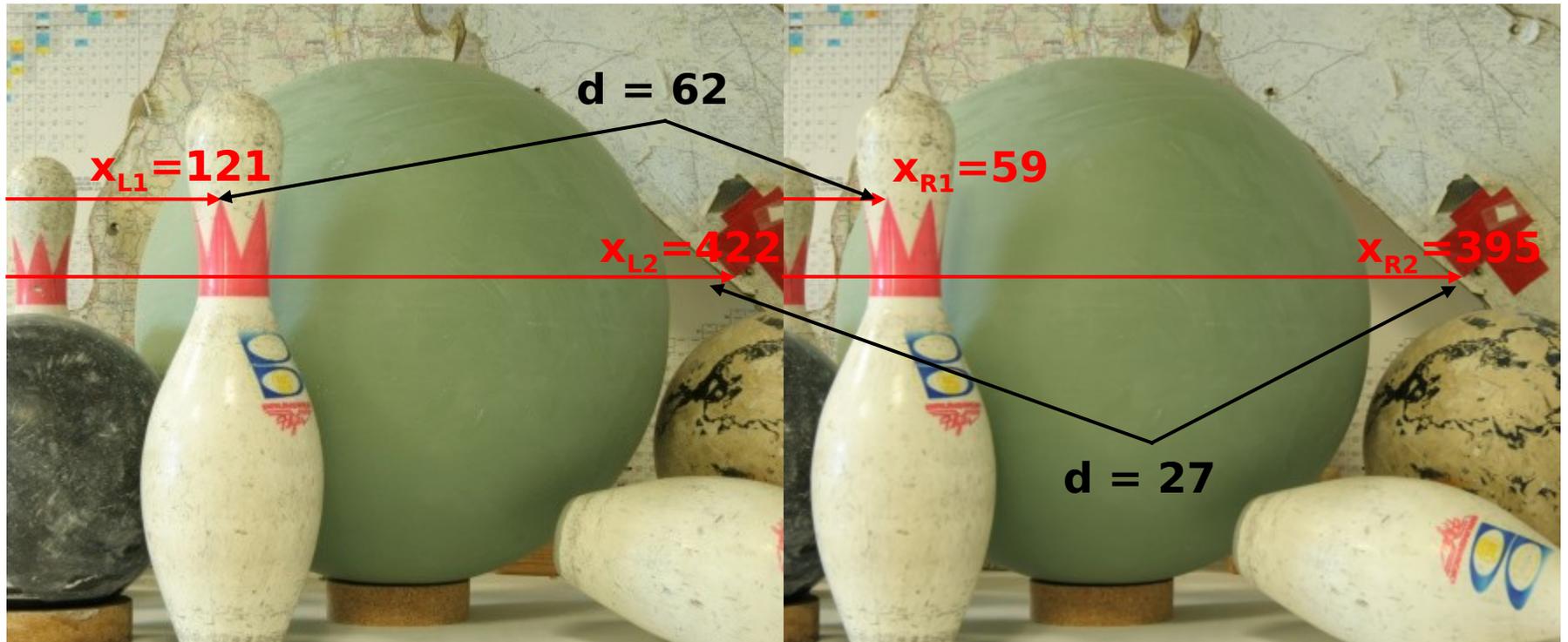


$$\frac{f}{x_l - x_r} = \frac{Z}{T}$$

$$Z = \frac{fT}{d}$$

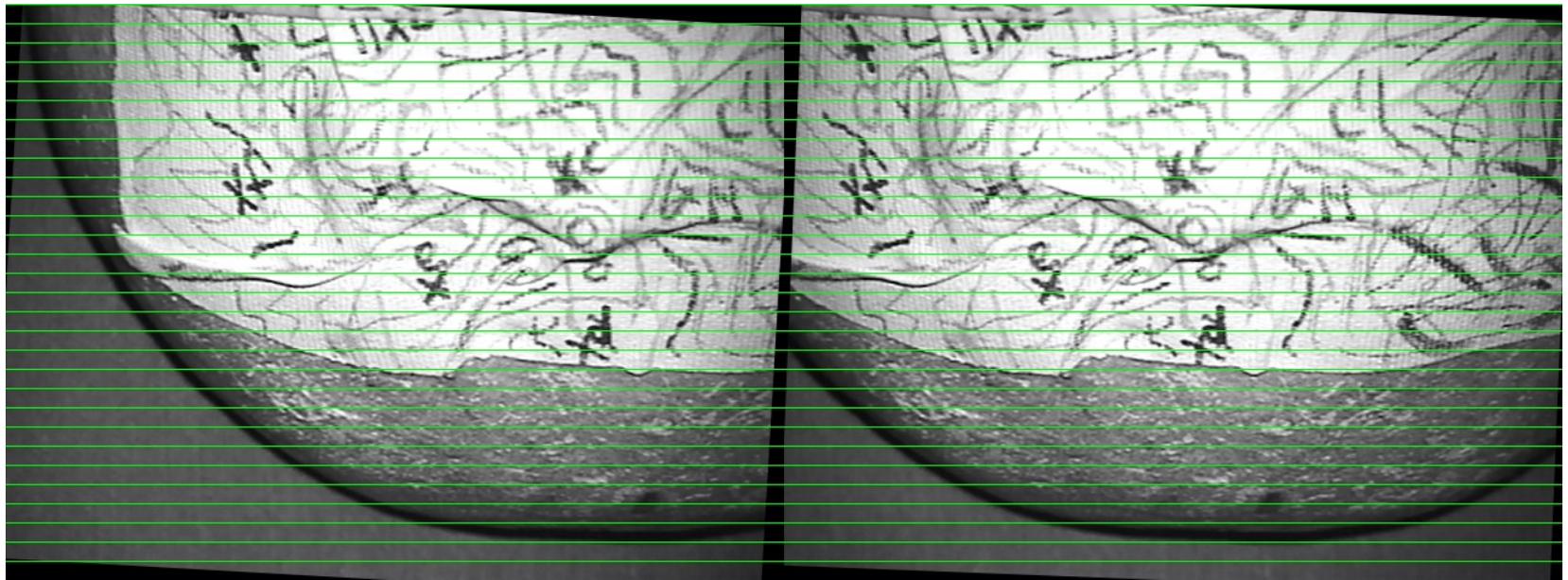
# RECONSTRUCCIÓN ESTÉREO

## Profundidad y disparidad



# RECONSTRUCCIÓN ESTÉREO

## Imágenes rectificadas



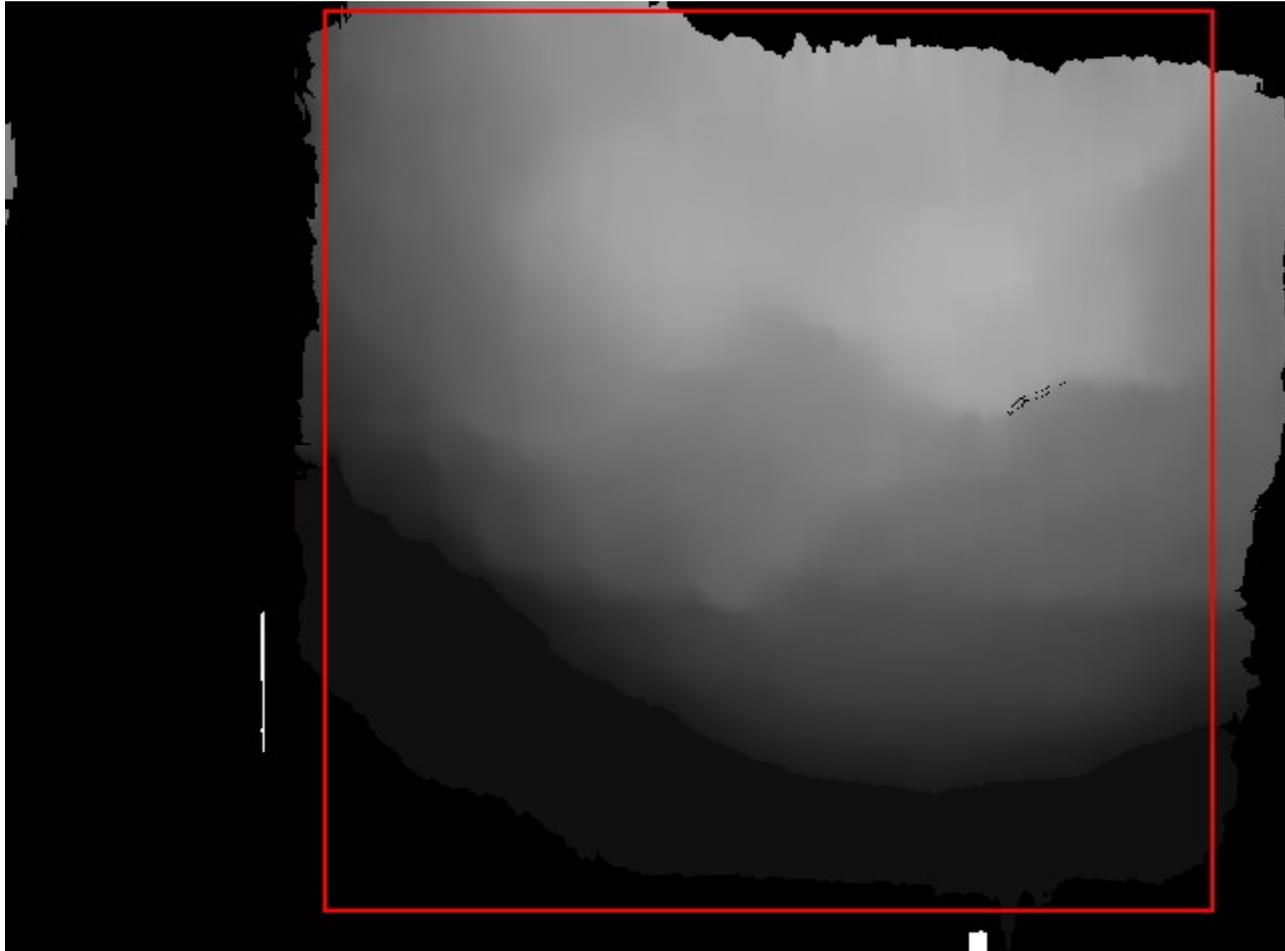
# RECONSTRUCCIÓN ESTÉREO

## Búsqueda de puntos correspondientes - Mapas de disparidad

- Block Matching
- Graph Cut

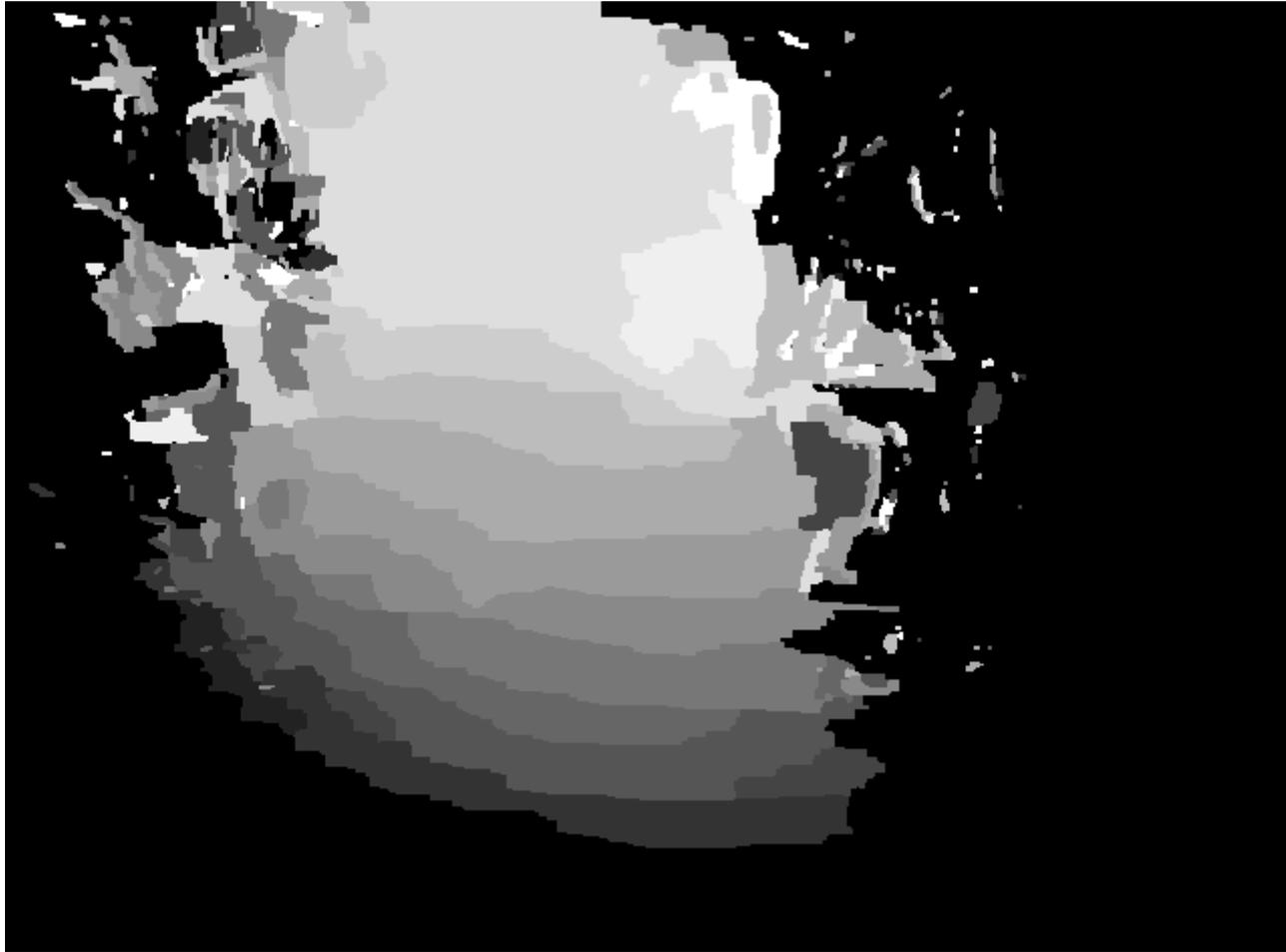
# RECONSTRUCCION ESTEREO

## Mapa de disparidad - Block Matching



# RECONSTRUCCIÓN ESTÉREO

## Mapa de disparidad - Graph Cut



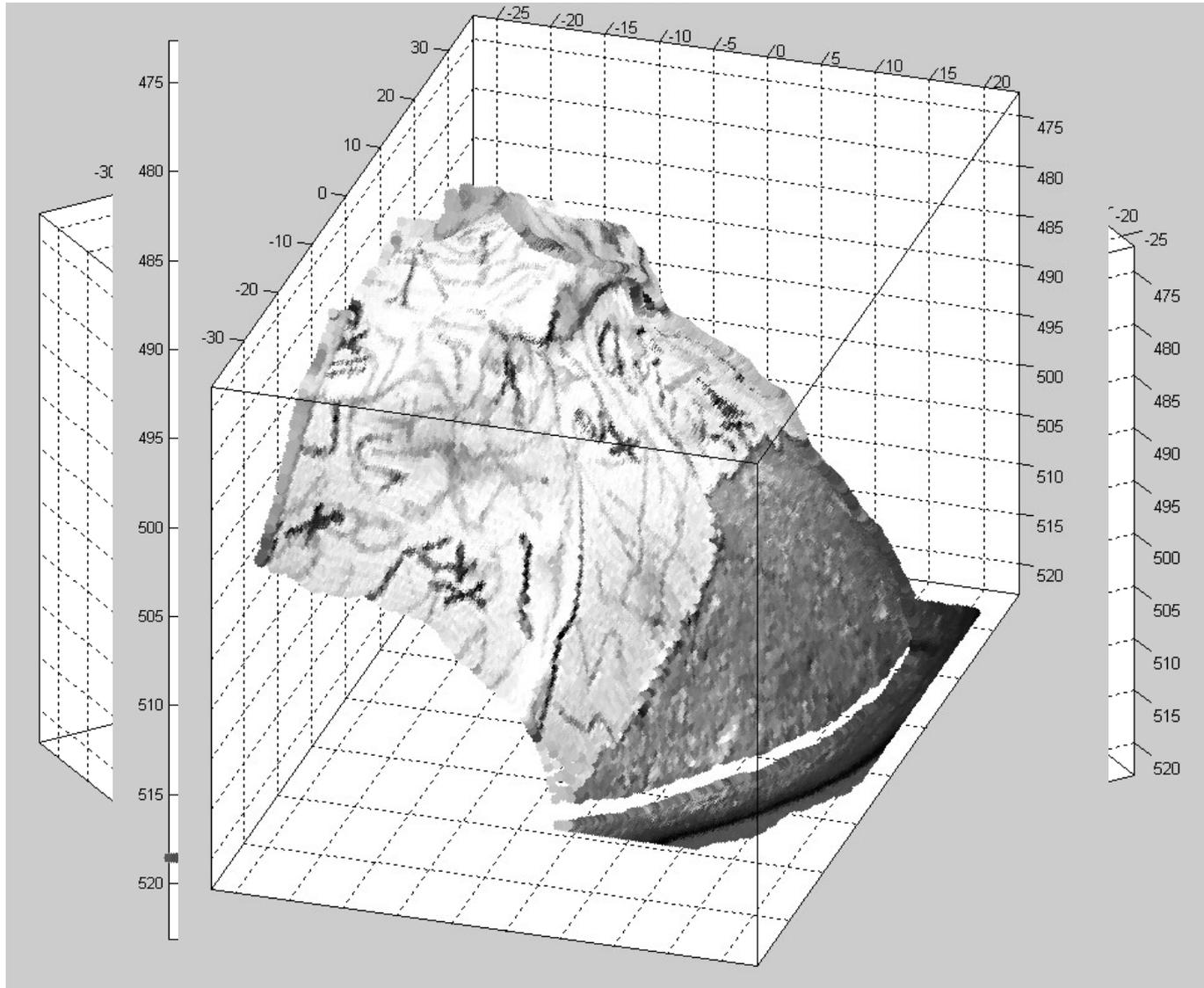
# RECONSTRUCCIÓN ESTÉREO

Reconstrucción utilizando Graph Cut



# RECONSTRUCCIÓN ESTÉREO

## Mapeo de textura



# Reconstrucción Estéreo

## Ejemplo Tesla

Problem: Per-Camera Detection Then Fusion

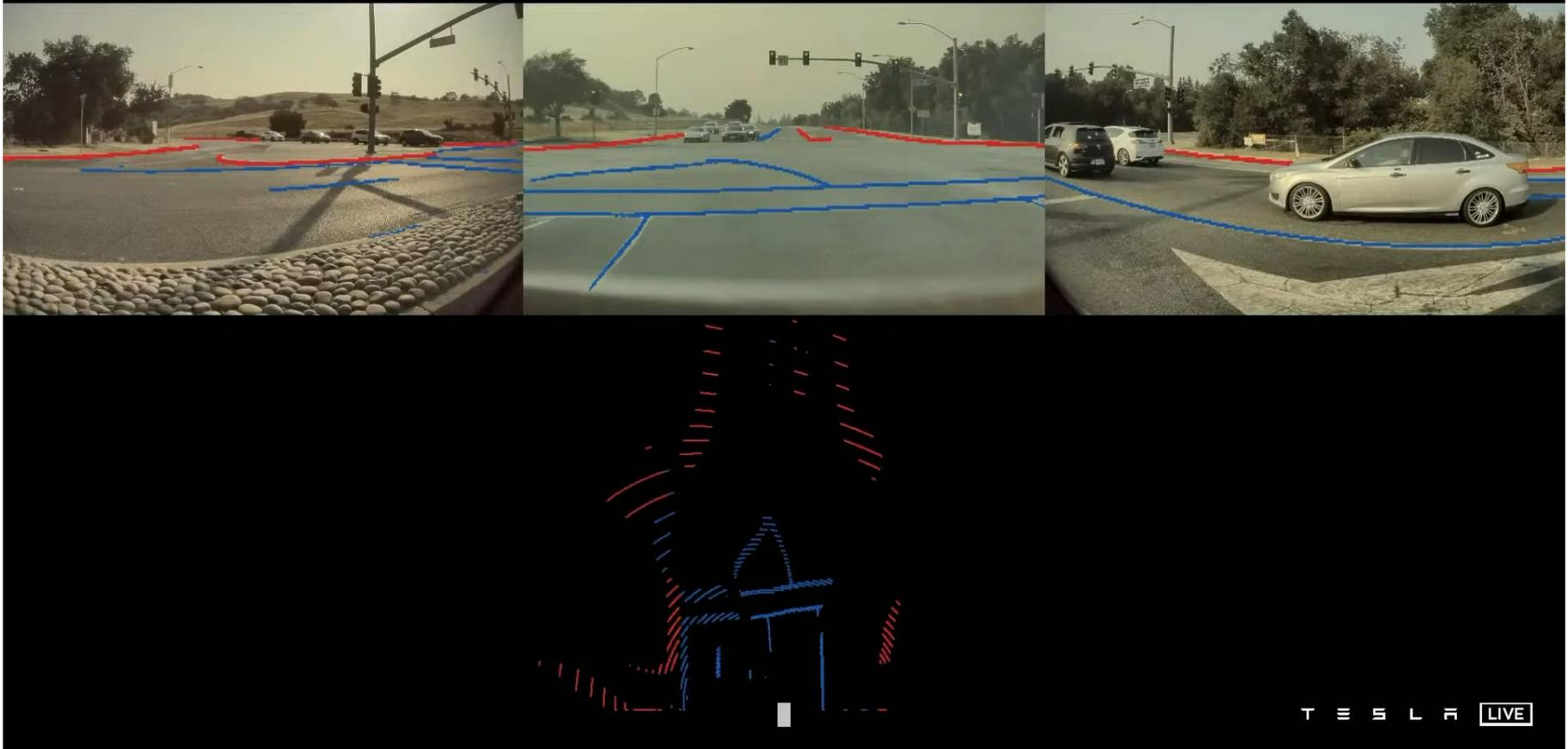


Figura extraída de presentación de [Tesla AI Day](#)

# Referencias

- Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.
- Zhang, Zhengyou. "A flexible new technique for camera calibration." IEEE Transactions on pattern analysis and machine intelligence 22.11 (2000): 1330-1334.
- **Camera Calibration Tollbox For Matlab**