

VISIÓN 3D

Reconstrucción 3D



Ingeniería de Sistemas y Automática

Universidad Miguel Hernández

Tabla de Contenidos

- **Reconstrucción 3D**
 - Ejes Alineados
 - Ejes Convergentes
 - Mapa de Disparidad
- Reconstrucción Proyectiva (Autocalibración)
- Odometría Visual

Reconstrucción 3D

- **OBJETIVO:** conocer las coordenadas de un punto del espacio:

$$(x_w, y_w, z_w) \equiv (X, Y, Z)$$

- **DATOS:**

- Suponemos conocidas la proyección del punto (X, Y, Z) sobre el plano imagen de las dos cámaras
- A partir de las coordenadas laterales de los pixeles obtenemos las coordenadas sin distorsión.

Pixel cámara izquierda $(x_f^l, y_f^l) \Rightarrow (x_n^l, y_n^l) \Rightarrow (x, y) \Rightarrow \tilde{m}_u$

Pixel cámara derecha $(x_f^r, y_f^r) \Rightarrow (x_n^r, y_n^r) \Rightarrow (x', y') \Rightarrow \tilde{m}'_u$

- **DATOS DE CALIBRACIÓN :**

$$\tilde{m}_u \cong A [I \quad 0] \tilde{M}_c$$

$$\tilde{m}'_u \cong A' [R \quad t] \tilde{M}_c$$

Distorsión: $\begin{bmatrix} x_n^d \\ y_n^d \\ 1 \end{bmatrix} \cong A^{-1} \cdot \tilde{m} \quad \longrightarrow \quad \tilde{m}_u \cong A \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$

↑

$$\left. \begin{aligned} x_n^d &= x_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_n y_n + p_2(2x_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ y_n^d &= y_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x_n y_n + p_1(2y_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ r^2 &= x_n^2 + y_n^2 \end{aligned} \right\}$$

Reconstrucción 3D



Imagen izquierda



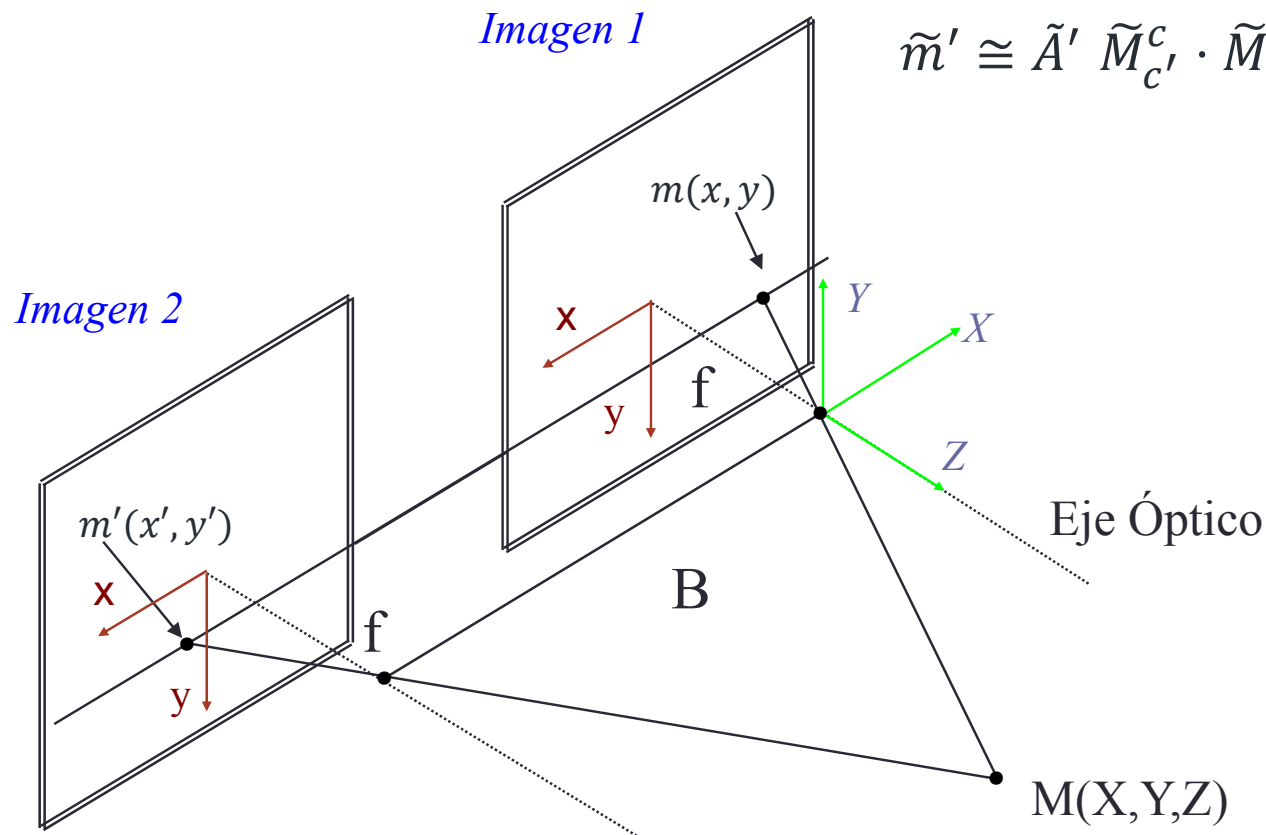
Imagen derecha

Reconstrucción 3D. Ejes Alineados

- Ejes Alineados
 - Sistema de coordenadas 3D referido a la cámara izquierda
 - Sistemas de coordenadas 2D: coordenadas centrales normalizadas ($f=f'$)

$$\tilde{m} \cong \tilde{A} \cdot \tilde{M}_c = [A \quad 0] \begin{bmatrix} M \\ 1 \end{bmatrix} \quad \tilde{M}_{c'}^c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$\tilde{m}' \cong \tilde{A}' \tilde{M}_{c'}^c \cdot \tilde{M}_c = A' [R \quad t] \begin{bmatrix} M \\ 1 \end{bmatrix}$$



$$\tilde{m} \cong A M$$

$$\tilde{m}' \cong A' R M + A' t$$

$$R = I$$

$$A = A' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad t = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}$$

Reconstrucción 3D. Ejes Alineados

- Ecuaciones de reconstrucción:

$$\tilde{m} \cong A M$$

$$R = I \quad t = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}$$

$$A = A' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{m}' \cong A' RM + A't$$

- Cámara izquierda:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow \left. \begin{array}{l} \lambda x = f X \\ \lambda y = f Y \\ \lambda = Z \end{array} \right\} \Rightarrow$$

$$\begin{array}{l} X = \frac{Z x}{f} \\ Y = \frac{Z y}{f} \end{array}$$

$$\begin{array}{l} Z = \frac{f \cdot B}{(x' - x)} \\ y' = y \end{array}$$

- Cámara Derecha:

$$\begin{bmatrix} \mu x' \\ \mu y' \\ \mu \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \Rightarrow \left. \begin{array}{l} \mu x' = f X + f B \\ \mu y' = f Y \\ \mu = Z \end{array} \right\} \Rightarrow \begin{array}{l} Z x' = f \frac{Z x}{f} + f B \\ Z y' = f \frac{Z y}{f} \end{array}$$

Reconstrucción 3D. Ejes Alineados

- Ecuaciones de Reconstrucción

- Coordenadas de un punto en la Imagen 1 (izquierda)

$$(x_f^l, y_f^l) \Rightarrow (x_u^l, y_u^l) \equiv (x, y) = m$$

- Coordenadas de un punto en la Imagen 2 (derecha)

$$(x_f^r, y_f^r) \Rightarrow (x_u^r, y_u^r) \equiv (x', y') = m'$$

- Disparidad: $d = x' - x$ al ser $y' = y$

$$X = \frac{B \cdot x}{d}$$

$$Y = \frac{B \cdot y}{d}$$

$$Z = \frac{f \cdot B}{d}$$

Disparidad

- Existe una relación inversa entre Z y d . Para unos determinados valores de B, f : Puntos con igual disparidad tienen la misma Z
- Para unos determinados valores de Z, f : Cuanto mayor sea la B , mayor será la d , y menos influirán los errores en la estimación de las coordenadas de los puntos
- Para valores grandes de B es más difícil encontrar la correspondencia de un punto en las dos imágenes.

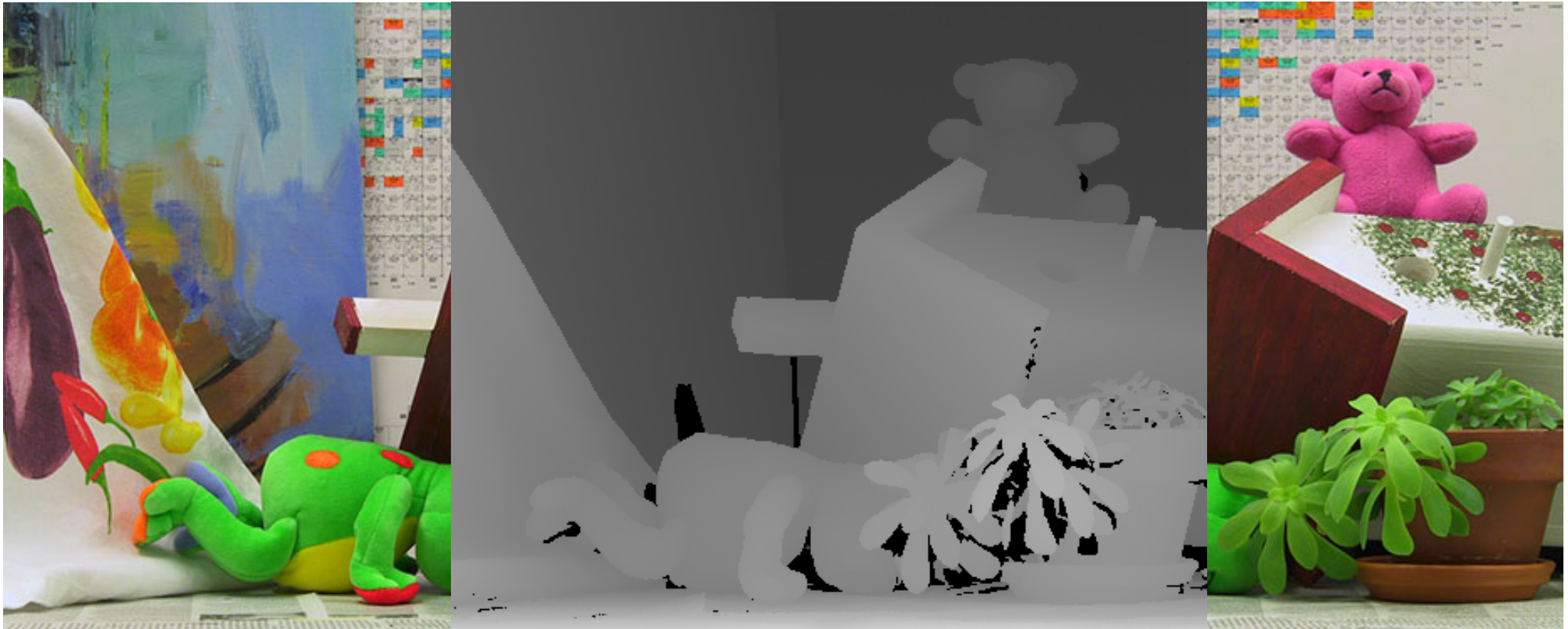
$$X = \frac{B \cdot x}{d}$$

$$Y = \frac{B \cdot y}{d}$$

$$Z = \frac{f \cdot B}{d}$$

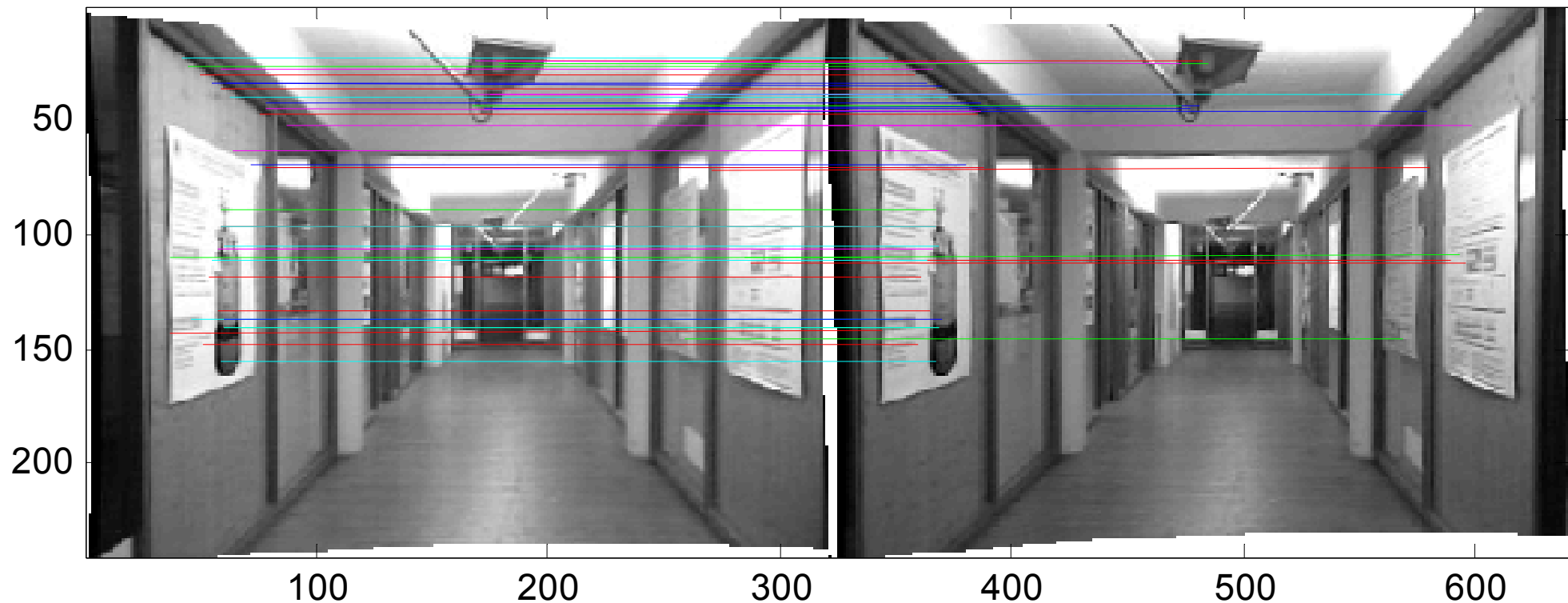
Mapa de Disparidad

- El mapa de disparidad es inversamente proporcional a la profundidad de los puntos en la escena



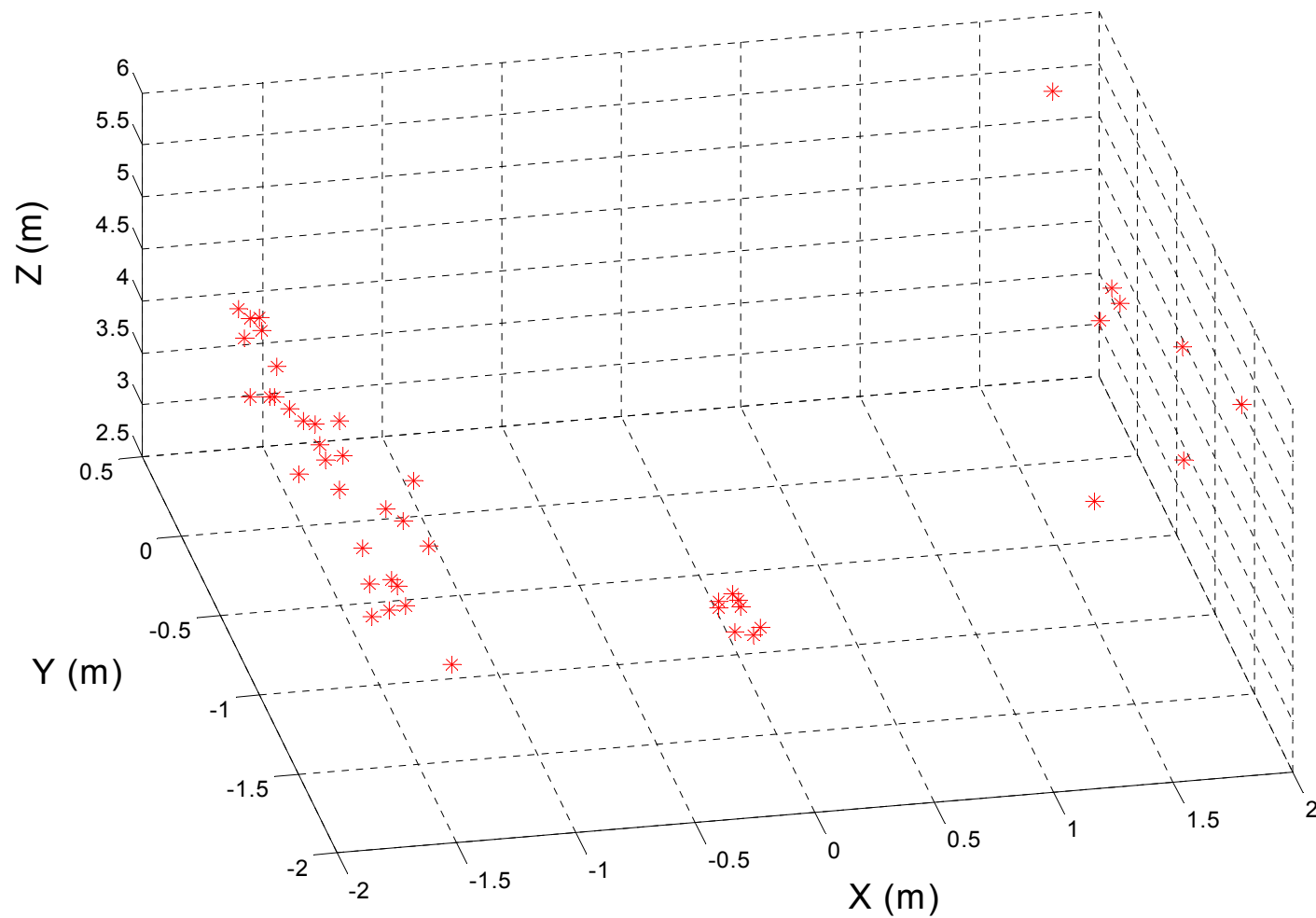
Ejemplo: Reconstrucción estéreo dispersa

- Correspondencias halladas. Restricciones:
 - Epipolar: ± 1 pixel
 - Descriptor SIFT de cada punto



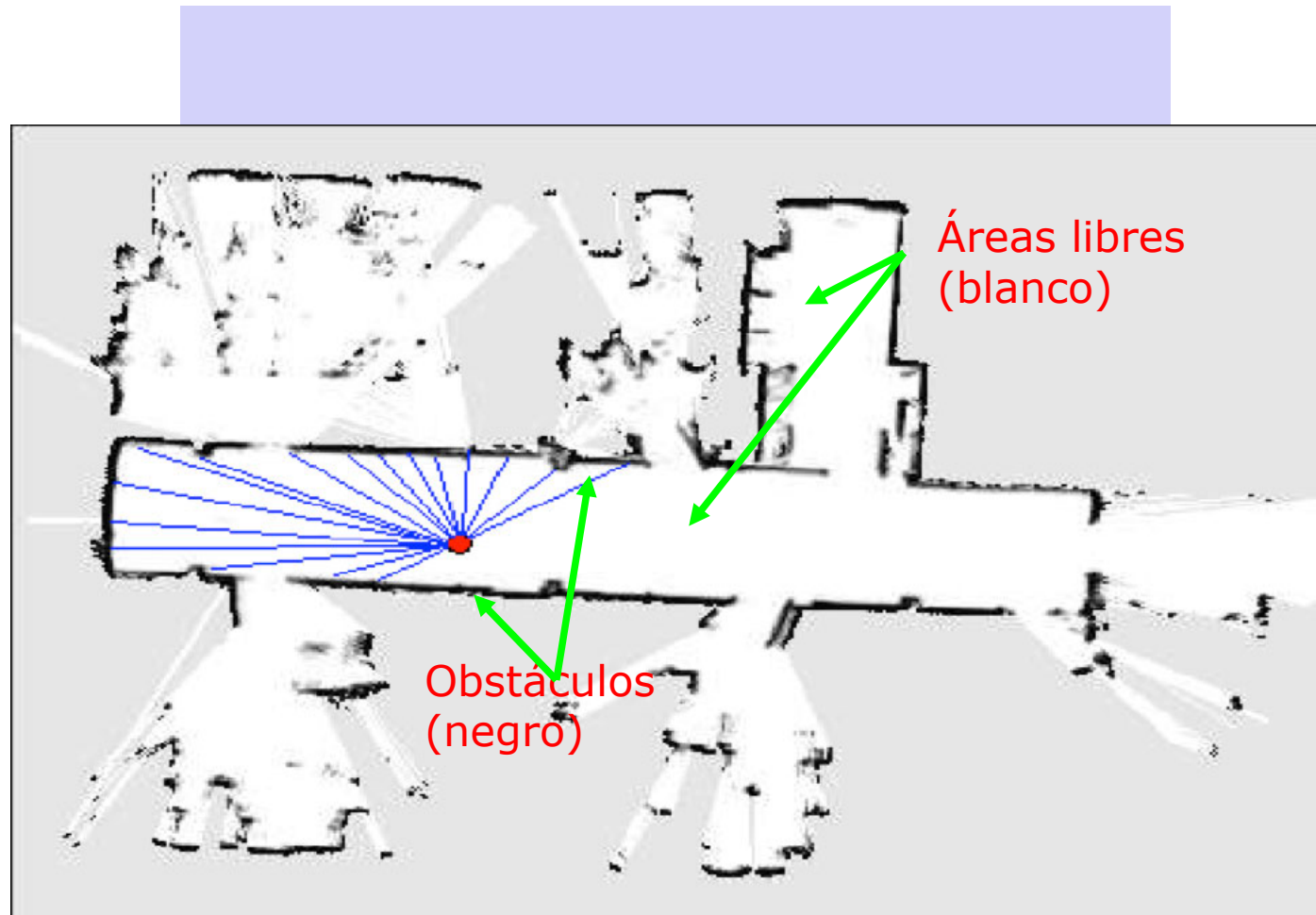
Ejemplo: SIFT

- Reconstrucción 3D



Reconstrucción 3D

- Robótica móvil: creación de mapas



Video

Precisión

- Nos interesa conocer la precisión de la medida en función de los parámetros del par estéreo.
- Vamos a buscar una expresión aproximada del error en la estimación de la profundidad:

$$Z = f \frac{B}{d}$$

$$\varepsilon_z = \left| \frac{\partial Z}{\partial B} \right| |\varepsilon_B| + \left| \frac{\partial Z}{\partial d} \right| |\varepsilon_d| + \left| \frac{\partial Z}{\partial f} \right| |\varepsilon_f| =$$

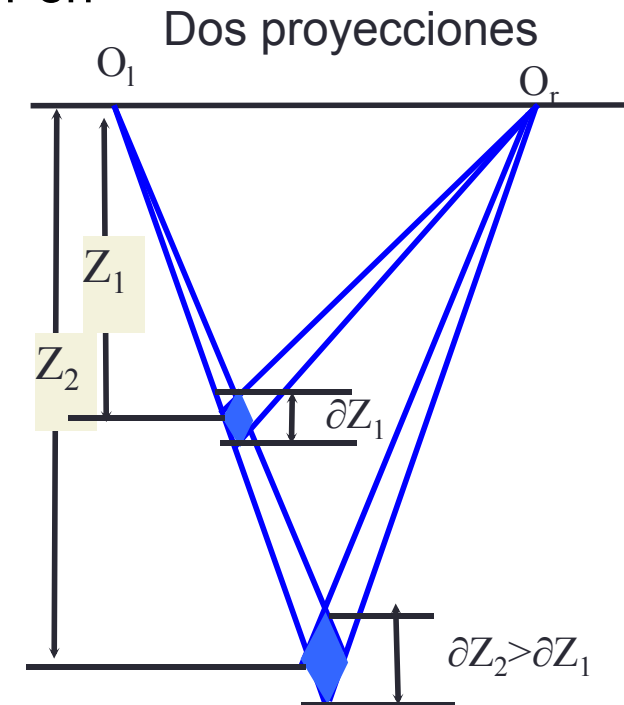
$$\left| \frac{f}{d} \right| |\varepsilon_B| + \left| \frac{fB}{d^2} \right| |\varepsilon_d| + \left| \frac{B}{d} \right| |\varepsilon_f|$$

- Nos interesa el error en función de la disparidad. Podemos considerar que:

$$\varepsilon_z \approx \left| \frac{fB}{d^2} \right| |\varepsilon_d|$$

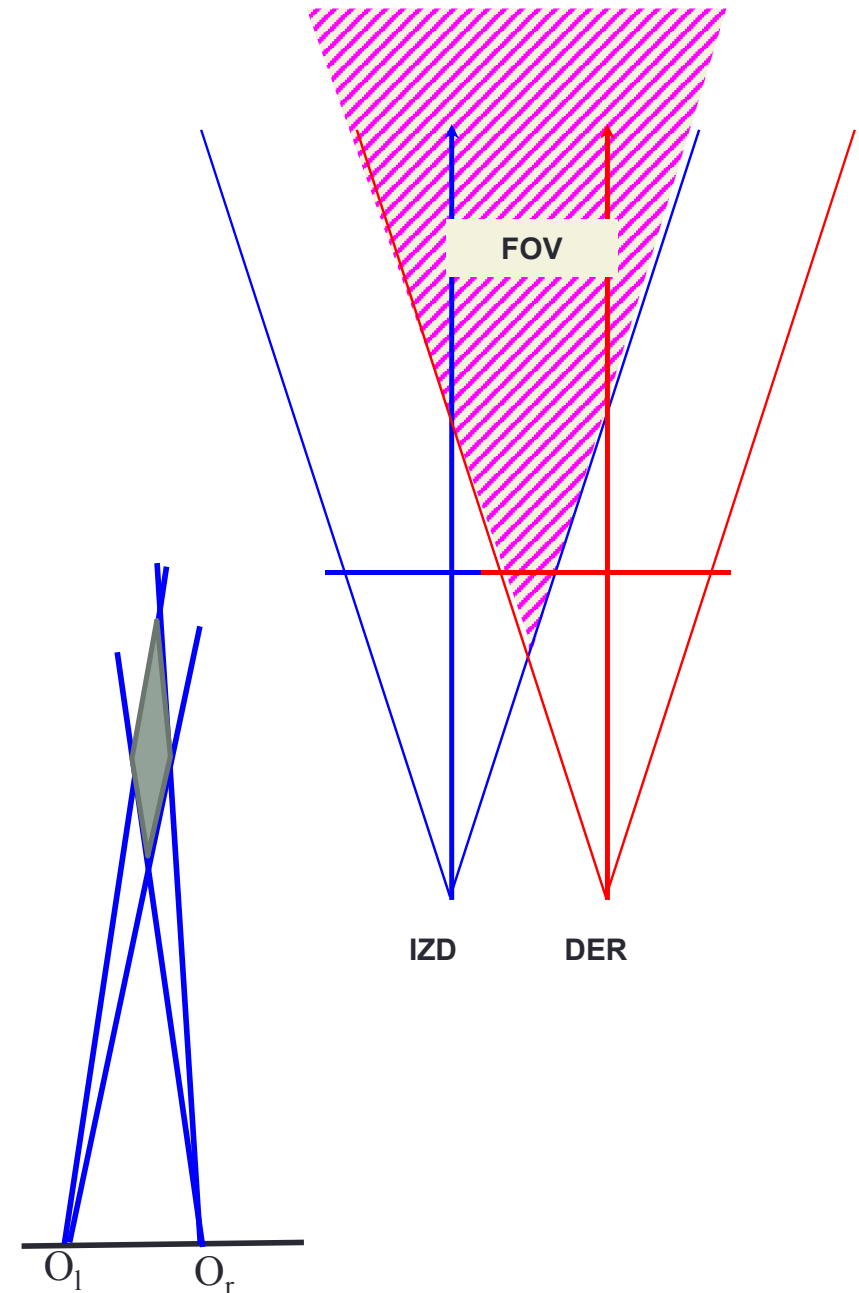
$$|\varepsilon_d| = 1 \text{ pixel}$$

- Si **Z pequeña** → **d grande**, cometemos menos error en la estimación.
- Al limitar la disparidad mínima, limitamos el error absoluto máximo que cometemos en la reconstrucción 3D.



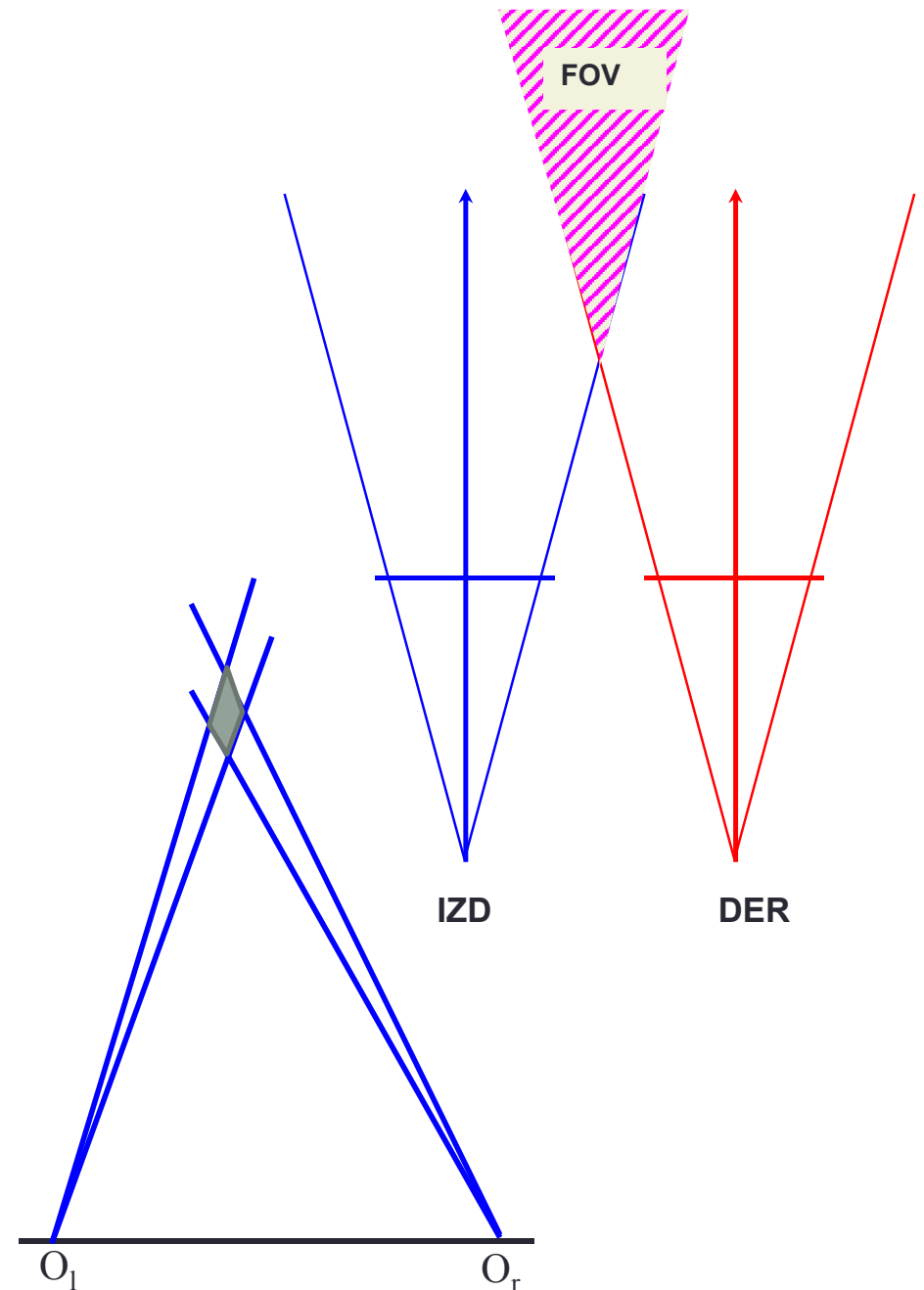
Precisión

- **B** pequeña
 - Ventajas:
 - FOV común grande
 - Aumenta la cantidad de posibles correspondencias. Imágenes similares
 - Inconvenientes:
 - Mayor error en la estimación de Z
 - Disparidad pequeña para puntos correspondientes $\rightarrow$ poca precisión
 - Puntos alejados tienen disparidad $\rightarrow 0$



Precisión

- **B grande**
 - Ventajas
 - Mejor precisión en la estimación de la profundidad.
 - Inconvenientes
 - FOV común pequeño
 - Imágenes muy diferentes
- Alternativas
 - Si el entorno es conocido seleccionar B para que el FOV ocupe toda la escena.
 - Ejes convergentes
- Ojo! FOV: no es el de cada cámara. Nos referimos al FOV común.



Precisión: línea Base (B)

- Reconstrucción con dos cámaras de ejes alineado

$$Z = \frac{f B}{x_2 - x_1} = \frac{f B}{d}$$

- Datos: $K_x=30\text{pixel/mm}$; $Z=2000\text{mm}$ $f=20\text{mm}$; *Error detección = $\pm 2\text{pixel}$*

$$\Rightarrow B = 100\text{mm} \quad d = \frac{fB}{Z} = \frac{20 \cdot 100}{2000} = 1.0\text{mm} \Rightarrow 30\text{píxeles}$$

$$\begin{cases} d = 28\text{píxeles} \Rightarrow d = 0.9333\text{mm} & Z = \frac{fB}{d} = \frac{20 \cdot 100}{0.9333} = 2142.93 \\ d = 32\text{píxeles} \Rightarrow d = 1.0666\text{mm} & Z = \frac{fB}{d} = \frac{20 \cdot 100}{1.0666} = 1875.11 \end{cases}$$

$$\text{Rango de detección: } 2142.93 - 1875.11 = 267.82$$

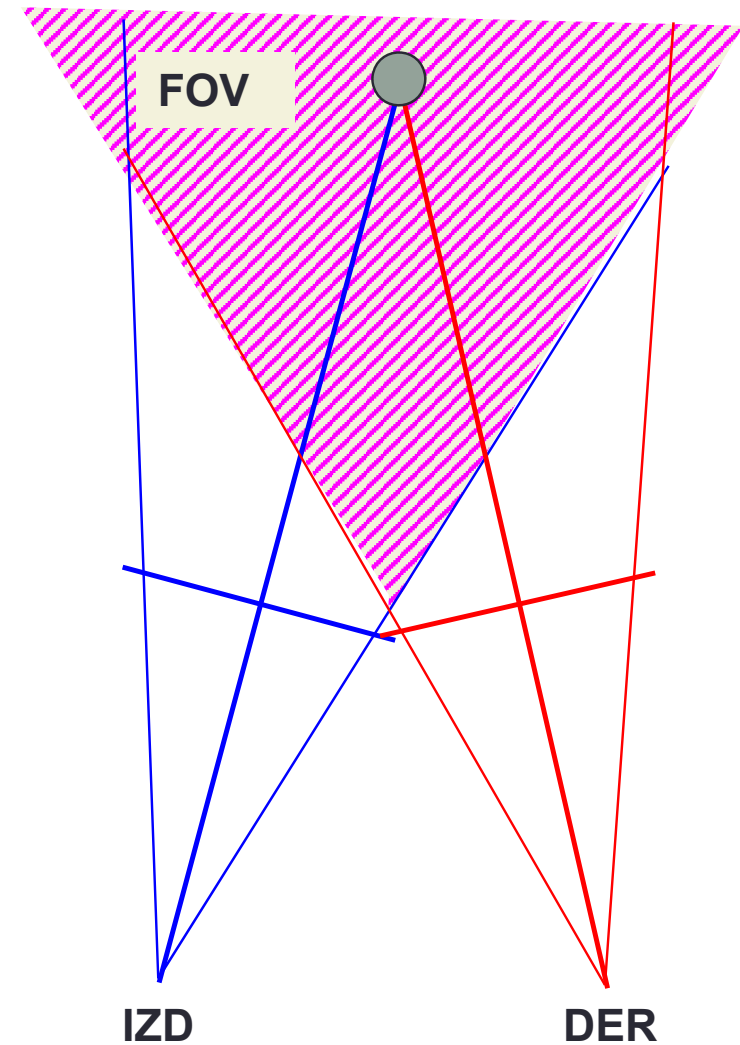
$$\Rightarrow B = 500\text{mm} \quad d = \frac{fB}{Z} = \frac{20 \cdot 500}{2000} = 5.0\text{mm} \Rightarrow 150\text{píxeles}$$

$$\begin{cases} d = 148\text{píxeles} \Rightarrow d = 4.9333\text{mm} & Z = \frac{fB}{d} = \frac{20 \cdot 500}{4.9333} = 2027.04 \\ d = 152\text{píxeles} \Rightarrow d = 5.0666\text{mm} & Z = \frac{fB}{d} = \frac{20 \cdot 500}{5.0666} = 1973.71 \end{cases}$$

$$\text{Rango de detección: } 2027.04 - 1973.71 = 53.33$$

Precisión: Estéreo con ejes convergentes

- Ejes convergentes:
 - FOV mayor.
 - Búsqueda de correspondencias en las líneas epipolares.
 - Existe un punto a distancia Z finita con disparidad cero.



Reconstrucción 3D. Ejes Convergentes

- **OBJETIVO:** conocer las coordenadas de un punto del espacio:

$$(x_w, y_w, z_w) \equiv (X, Y, Z)$$

- **DATOS:**

- Suponemos conocidas la proyección del punto (X, Y, Z) sobre el plano imagen de las dos cámaras
- A partir de las coordenadas laterales de los pixeles obtenemos las coordenadas sin distorsión.

Pixel cámara izquierda $(x_f^l, y_f^l) \Rightarrow (x_n^l, y_n^l) \Rightarrow (x, y) \Rightarrow \tilde{m}_u$

Pixel cámara derecha $(x_f^r, y_f^r) \Rightarrow (x_n^r, y_n^r) \Rightarrow (x', y') \Rightarrow \tilde{m}'_u$

- **DATOS DE CALIBRACIÓN :**

$$\tilde{m}_u \cong A [I \quad 0] \tilde{M}_c$$

$$\tilde{m}'_u \cong A' [R \quad t] \tilde{M}_c$$

Distorsión: $\begin{bmatrix} x_n^d \\ y_n^d \\ 1 \end{bmatrix} \cong A^{-1} \cdot \tilde{m} \quad \longrightarrow \quad \tilde{m}_u \cong A \cdot \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$

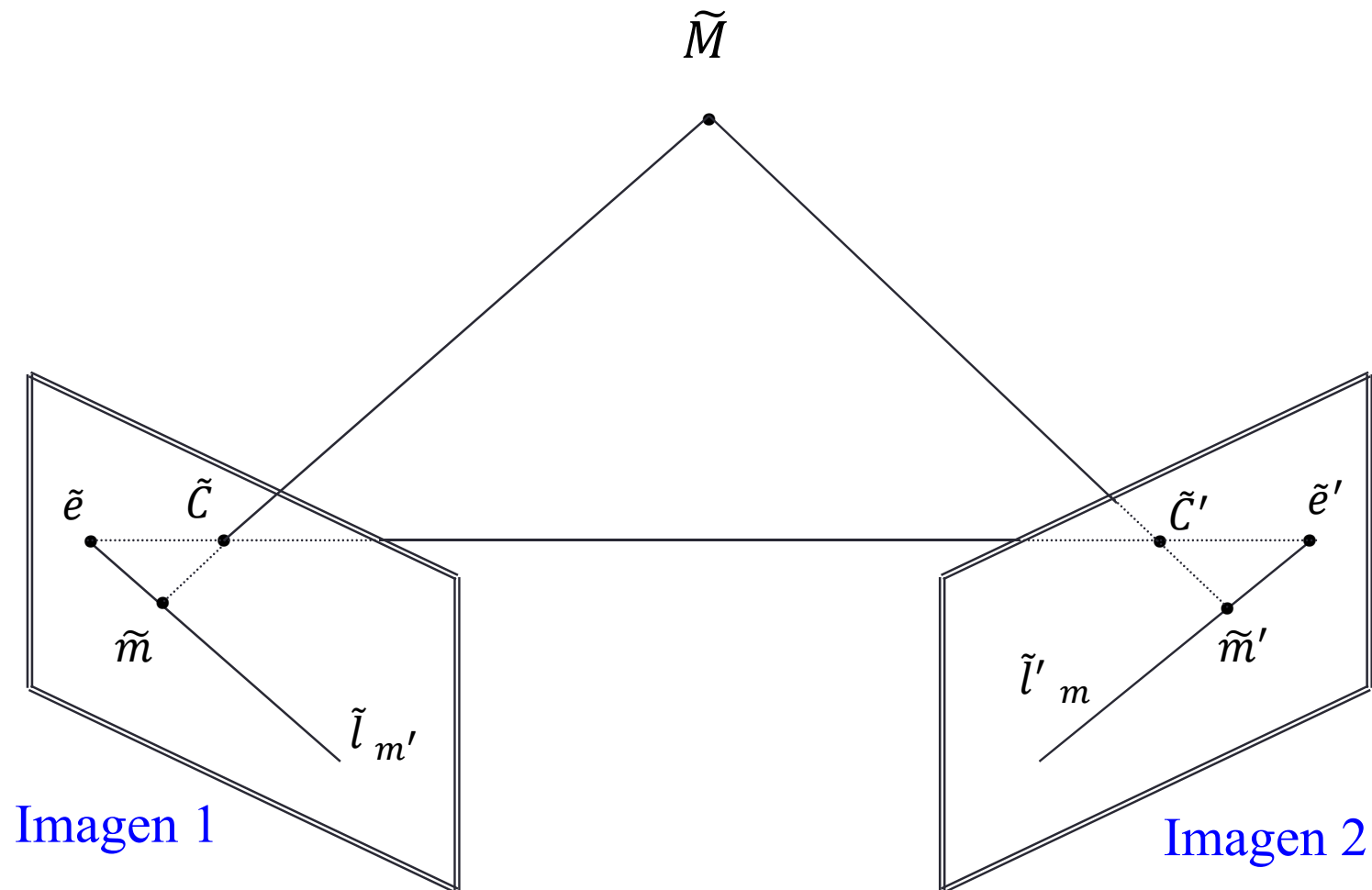
↑

$$\left. \begin{aligned} x_n^d &= x_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_n y_n + p_2(2x_n^2 + r^2) + s_1 r^2 + s_2 r^4 \\ y_n^d &= y_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x_n y_n + p_1(2y_n^2 + r^2) + s_1 r^2 + s_2 r^4 \end{aligned} \right\}$$

$$r^2 = x_n^2 + y_n^2$$

Reconstrucción 3D. Ejes Convergentes

- Triangulación



Triangulación (Ejes Convergentes)

- Rectas proyección:

$$\tilde{P} \cong A [I \ 0]$$

$$\tilde{P}' \cong A' [R \ t]$$

$$\tilde{m}_i \cong \tilde{P} \tilde{M}_i \Rightarrow \begin{bmatrix} \lambda \ x \\ \lambda \ y \\ \lambda \end{bmatrix} = \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \tilde{M}_i$$

$$\tilde{m}'_i \cong \tilde{P}' \tilde{M}_i \Rightarrow \begin{bmatrix} \mu x' \\ \mu y' \\ \mu \end{bmatrix} = \begin{bmatrix} P_1'^T \\ P_2'^T \\ P_3'^T \end{bmatrix} \tilde{M}_i$$

$$\begin{aligned} P_3^T \tilde{M}_i x &= P_1^T \tilde{M}_i \\ P_3^T \tilde{M}_i y &= P_2^T \tilde{M}_i \end{aligned} \Rightarrow \begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \end{bmatrix} \tilde{M}_i = 0$$

$$\begin{aligned} P_3'^T \tilde{M}_i x' &= P_1'^T \tilde{M}_i \\ P_3'^T \tilde{M}_i y' &= P_2'^T \tilde{M}_i \end{aligned} \Rightarrow \begin{bmatrix} P_3'^T x - P_1'^T \\ P_3'^T y - P_2'^T \end{bmatrix} \tilde{M}_i = 0$$

- Triangulación:
$$\begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \\ P_3'^T x' - P_1'^T \\ P_3'^T y' - P_2'^T \end{bmatrix} \tilde{M}_i = 0$$

$$A_{xy} \tilde{M}_i = 0$$

$$\min_{\tilde{M}} [(|A_{xy} \tilde{M}|)^2]$$

- Estimación mínimos cuadrados:

$$\begin{aligned} SVD(A_{xy}) &\rightarrow A_{xy} = UDV^T \\ \tilde{M} &= v_n \quad \|\tilde{M}\| = 1 \end{aligned}$$

- Estimación no lineal: error de re-proyección

$$\min_{\tilde{M}_i} [(\tilde{m} - \lambda^{-1} P \tilde{M}_i)^2 + (\tilde{m}' - \lambda'^{-1} P' \tilde{M}_i)^2]$$

Triangulación

- Triangulación (solución no homogénea - Euclídea):

$$\begin{cases} (p_{31}x - p_{11})X_w + (p_{32}x - p_{12})Y_w + (p_{33}x - p_{13})Z_w = p_{14} - p_{34}x \\ (p_{31}y - p_{21})X_w + (p_{32}y - p_{22})Y_w + (p_{33}y - p_{23})Z_w = p_{24} - p_{34}y \\ (p'_{31}x' - p'_{11})X_w + (p'_{32}x' - p'_{12})Y_w + (p'_{33}x' - p'_{13})Z_w = p'_{14} - p'_{34}x' \\ (p'_{31}y' - p'_{21})X_w + (p'_{32}y' - p'_{22})Y_w + (p'_{33}y' - p'_{23})Z_w = p'_{24} - p'_{34}y' \end{cases}$$

$$A_{xy}M = b$$

$$A_{xy} = \begin{bmatrix} (p_{31}x - p_{11}) & (p_{32}x - p_{12}) & (p_{33}x - p_{13}) \\ (p_{31}y - p_{21}) & (p_{32}y - p_{22}) & (p_{33}y - p_{23}) \\ (p'_{31}x' - p'_{11}) & (p'_{32}x' - p'_{12}) & (p'_{33}x' - p'_{13}) \\ (p'_{31}y' - p'_{21}) & (p'_{32}y' - p'_{22}) & (p'_{33}y' - p'_{23}) \end{bmatrix} \quad b = \begin{bmatrix} p_{14} - p_{34}x \\ p_{24} - p_{34}y \\ p'_{14} - p'_{34}x' \\ p'_{24} - p'_{34}y' \end{bmatrix}$$

- Estimación mínimos cuadrados (pseudo-inversa): $\min_M \left[(A_{xy}M - b)^2 \right]$

$$M = (A_{xy}^T A_{xy})^{-1} A_{xy}^T \cdot b$$

Comparación

- **EJES ALINEADOS**

- Ventaja: Simplicidad de cálculo
- Inconvenientes:
 - Dificultad de construcción
 - Si B es alto, las imágenes son muy distintas, con posibles malas correspondencias
 - Si B es bajo, se obtiene mala precisión

- **EJES CONVERGENTES**

- Ventajas:
 - Facilidad de construcción
 - Correspondencia sencilla si la disparidad es baja
- Inconvenientes:
 - Complejidad de cálculo
 - Si B es alto, la correspondencia es difícil, oclusiones
 - Si B es bajo, se obtiene mala precisión

Tabla de Contenidos

- Reconstrucción 3D
- Reconstrucción Proyectiva (Autocalibración)
- Odometría Visual

Objetivos

- Hasta que punto podemos obtener información de la ubicación de objetos en la escena y de las cámaras a partir de dos vistas (**no calibradas**)
- **Datos:** correspondencias entre puntos proyectados en dos planos de imagen

$$\tilde{m}_i \cong \tilde{P} \quad \tilde{M}_i \cong [P \quad p] \begin{bmatrix} M_i \\ 1 \end{bmatrix} \quad \tilde{m}'_i \cong \tilde{P}' \quad \tilde{M}_i \cong [P' \quad p'] \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

- Matriz Fundamental:

$$\tilde{m}'^T F \tilde{m} = 0$$

- Matriz Esencial:

$$\tilde{m} \cong A \tilde{q} \quad \tilde{m}' \cong A' \tilde{q}'$$

$$\tilde{q}'^T E \tilde{q} = 0$$

Esquema general

- Calcular la matriz Fundamental/Esencial a partir de la correspondencia de puntos (Calibración Proyectiva)

$$\tilde{m}_i \leftrightarrow \tilde{m}_i'$$

- Calcular las matrices proyectivas de cada cámara a partir de la matriz **Fundamental** o la matriz **Esencial** (conocemos los parámetros intrínsecos)
- **Triangulación**: para cada correspondencia $\tilde{m}_i \leftrightarrow \tilde{m}_i'$ calcular el punto en el espacio proyectivo que se proyecta en estos dos puntos.

Matrices de cada cámara P, P'

- A partir de la matriz fundamental (**F**):

$$F \cong A'^{-T} [t]_{\wedge} R A^{-1}$$

$$F \cong [e']_{\wedge} A' R A^{-1} \cong A'^{-T} R A^T [e]_{\wedge}$$

- Quedan determinadas salvo una transformación proyectiva del espacio 3D
- Una solución es (cámara canónica):

$$\tilde{P} \cong [I \quad | \quad 0] \quad \tilde{P}' \cong [[\tilde{e}'] \wedge F \quad | \quad \tilde{e}'] \quad \tilde{F}^T \cdot \tilde{e}' = 0$$

- P' es singular $\Rightarrow$ centro óptico en el plano del infinito

Modelo proyectivo estéreo:

$$\left. \begin{array}{l} \tilde{P} \cong A [I \quad | \quad 0] \\ \tilde{P}' \cong A' [R \quad | \quad t] \\ \tilde{e}' \cong A' t \end{array} \right\} \Rightarrow \left. \begin{array}{l} \tilde{P} \cong [I \quad | \quad 0] \\ \tilde{P}' \cong [A' R A^{-1} \quad | \quad A' t] \end{array} \right\} \Rightarrow \left. \begin{array}{l} \tilde{P} \cong [I \quad | \quad 0] \\ \tilde{P}' \cong [H_{\infty} \quad | \quad \lambda \tilde{e}'] \end{array} \right\}$$

Matrices de cada cámara P, P'

- A partir de la matriz Esencial (**E**): $E = [t]_{\wedge} R$
 - Quedan determinadas salvo un factor de escala y una ambigüedad de cuatro configuraciones posibles:

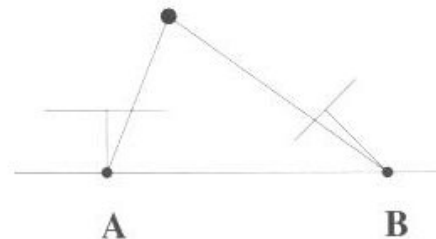
$$\tilde{P} = [I \mid 0]$$

$$\tilde{P}' = [R \mid t]$$

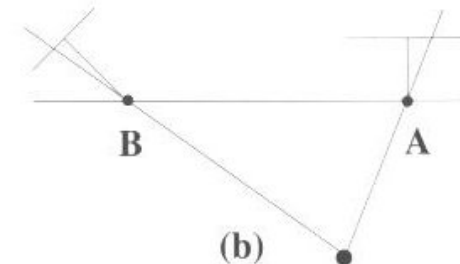
$$\tilde{P}' = \begin{bmatrix} UWV^T & |\lambda \cdot u_3| \\ UW^TV^T & |\lambda \cdot u_3| \\ UWV^T & |-\lambda \cdot u_3| \\ UW^TV^T & |-\lambda \cdot u_3| \end{bmatrix}$$

$$SVD(E) \rightarrow U \cdot D \cdot V^T$$

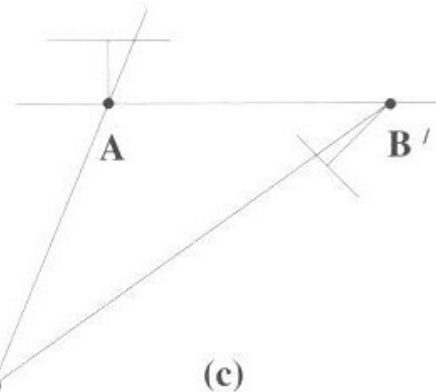
$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



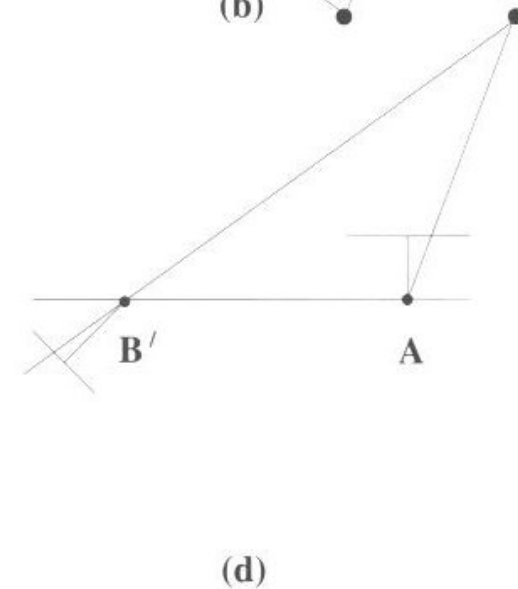
(a)



(b)



(c)



(d)

"A computer algorithm for reconstructing a scene from two projections"

H. Christopher Longuet-Higgins Nature. 293 (1981)

Triangulación

- Desambiguación a partir de la matriz Esencial:
 - Se reconstruyen los puntos 3D para las 4 posibles configuraciones
 - Nos quedamos con aquella en la que los puntos 3D están por delante de la cámara ($Z > 0$)

- Triangulación:
$$\begin{bmatrix} P_3^T x - P_1^T \\ P_3^T y - P_2^T \\ P_3'^T x' - P_1'^T \\ P_3'^T y' - P_2'^T \end{bmatrix} \tilde{M}_i = 0$$

- Estimación mínimos cuadrados:

$$\min_{\tilde{M}} \left[(|A_{xy} \tilde{M}|)^2 \right] \quad \begin{aligned} &SVD(A_{xy}) \rightarrow A_{xy} = UDV^T \\ &\tilde{M} = v_n \quad \|\tilde{M}\| = 1 \end{aligned}$$

“A computer algorithm for reconstructing a scene from two projections”

H. Christopher Longuet-Higgins Nature. 293 (1981)

Ambigüedad de la reconstrucción

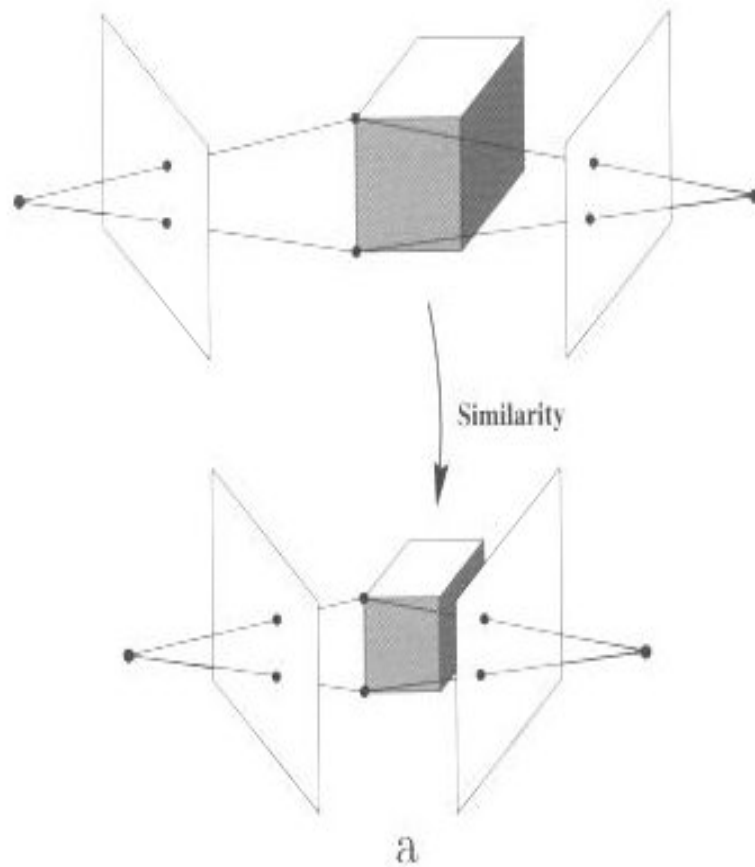
- De forma general no es posible reconstruir la posición y orientación absolutas respecto a un SC, sin información adicional acerca de la escena.
- Para cámaras no calibradas:
 - La escena queda determinada salvo una transformación proyectiva $\tilde{H}$
- Para un par de cámaras calibrado (A, A'):
 - La escena queda determinada salvo una transformación de similaridad (Rotación, Traslación y Escala)

$$\tilde{H}_S = \begin{bmatrix} R & t \\ 0 & \lambda \end{bmatrix}$$

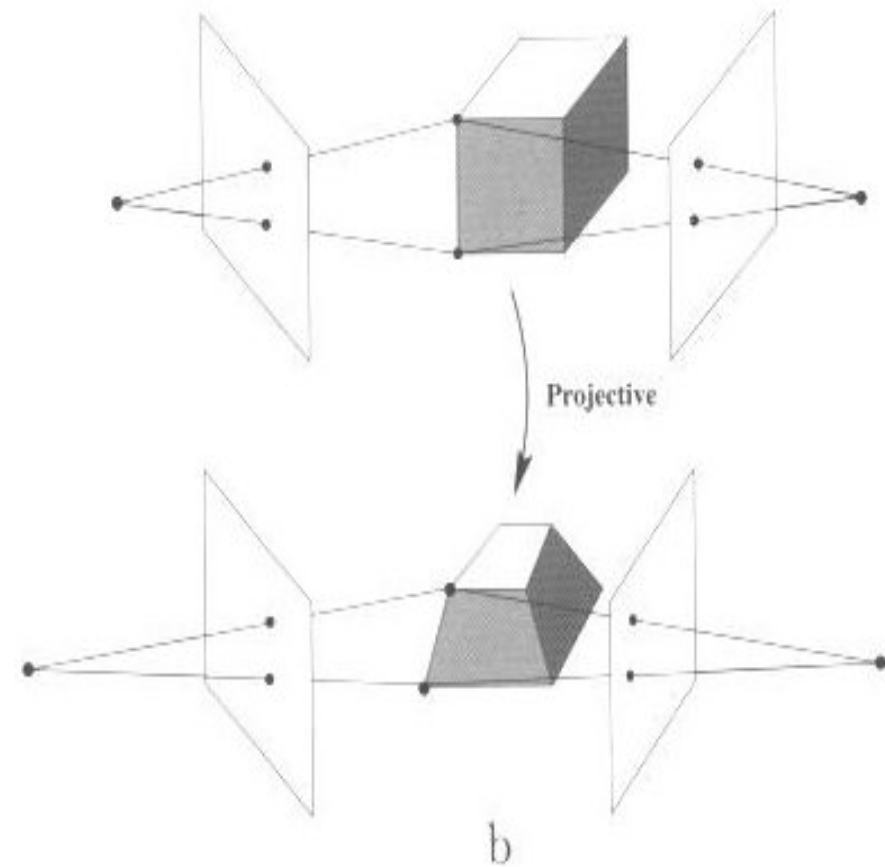
$$\text{Cambiando } \tilde{M}_i \rightarrow \tilde{H}_S \tilde{M}_i, \quad \tilde{P} \rightarrow \tilde{P} \tilde{H}_S^{-1}, \quad \tilde{P}' \rightarrow \tilde{P}' \tilde{H}_S^{-1}$$

$$\tilde{m}_i \cong \tilde{P} \tilde{M}_i \cong (\tilde{P} \tilde{H}_S^{-1})(\tilde{H}_S \tilde{M}_i), \quad \tilde{m}_i' \cong \tilde{P}' \tilde{M}_i' \cong (\tilde{P}' \tilde{H}_S^{-1})(\tilde{H}_S \tilde{M}_i')$$

Ambigüedad de la reconstrucción



Cámaras calibradas



Cámaras no calibradas

Reconstrucción Proyectiva

- Ventajas:
 - Permite reconstruir proyectivamente una escena **sin información acerca de la calibración** o posición de las dos cámaras.
 - La verdadera reconstrucción (**Euclidea**) esté en alguna transformación proyectiva de la anterior.

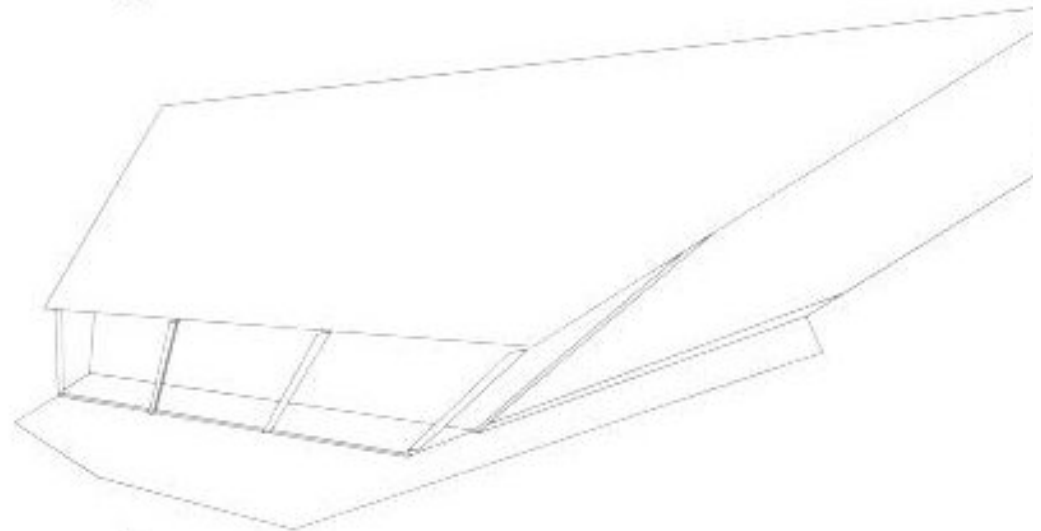
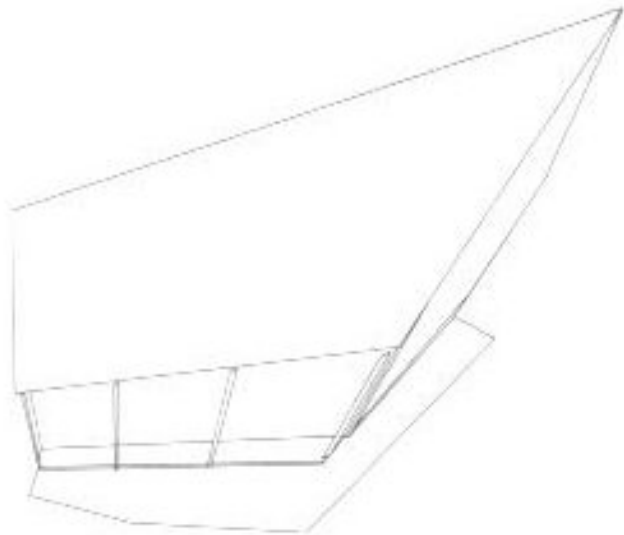
$$(\tilde{P}, \tilde{P}', \{\tilde{M}_i\}) \quad ; \quad (\tilde{P}_E, \tilde{P}_E', \{\tilde{M}_{Ei}\})$$

$$\tilde{P}_E = \tilde{P} \tilde{H}^{-1}, \quad \tilde{P}_E' = \tilde{P}' \tilde{H}^{-1}, \quad \tilde{M}_{Ei} = \tilde{H} \tilde{M}_i$$

Reconstrucción Proyectiva



a



b

*“Multiple View Geometry in Computer Vision” R. Hartley and A. Zisserman
Cambridge University Press, March 2004.*

Reconstrucción Proyectiva

- Aplicaciones:
 - Paralelaje: Navegación de robots, distancia colisión
 - Homografía entre vistas: modelado a partir de vistas

Paralelaje Proyectivo

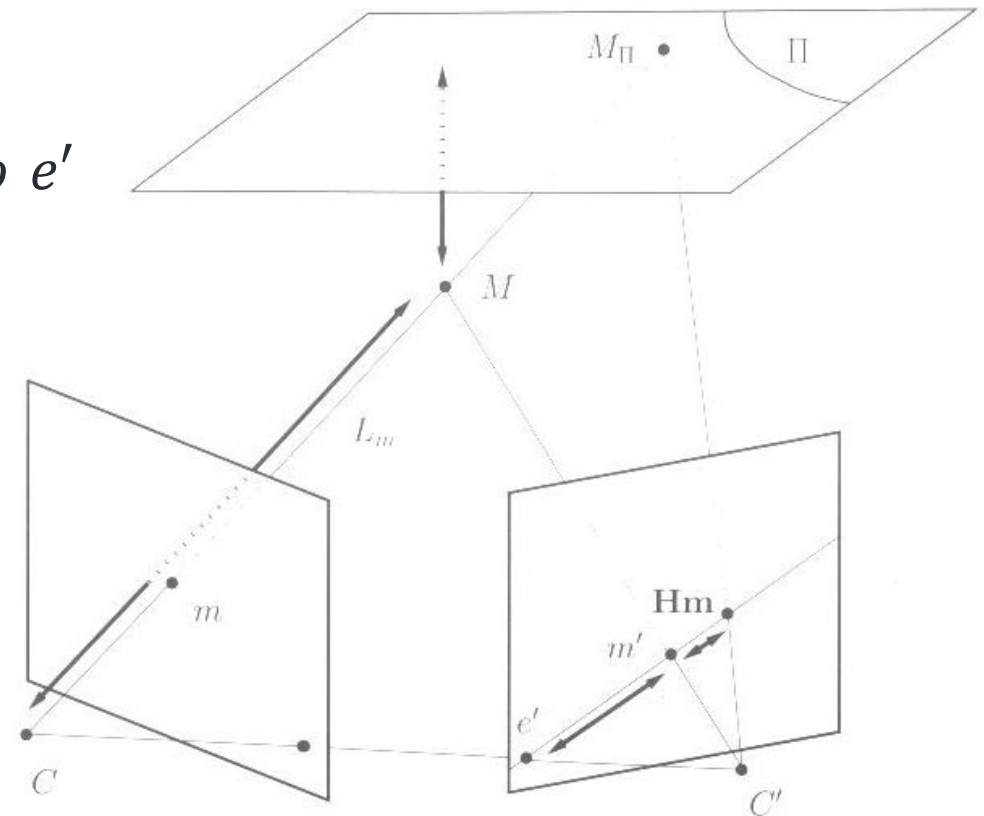
- **Paralelaje**: distancia a un plano (obstáculo)

$$\tilde{m} \cong [I_3 \quad | 0_3] \tilde{M} \Rightarrow \tilde{M} \cong \begin{bmatrix} \tilde{m} \\ \rho \end{bmatrix}$$

$$\tilde{m}' \cong [H \quad | \mu e'] \tilde{M} \Rightarrow \tilde{m}' \cong H\tilde{m} + \mu\rho e'$$

$$\tilde{m}' \cong H\tilde{m} + \kappa e'$$

$$[\tilde{m}']_{\wedge} H\tilde{m} + \kappa [\tilde{m}']_{\wedge} e' = 0$$



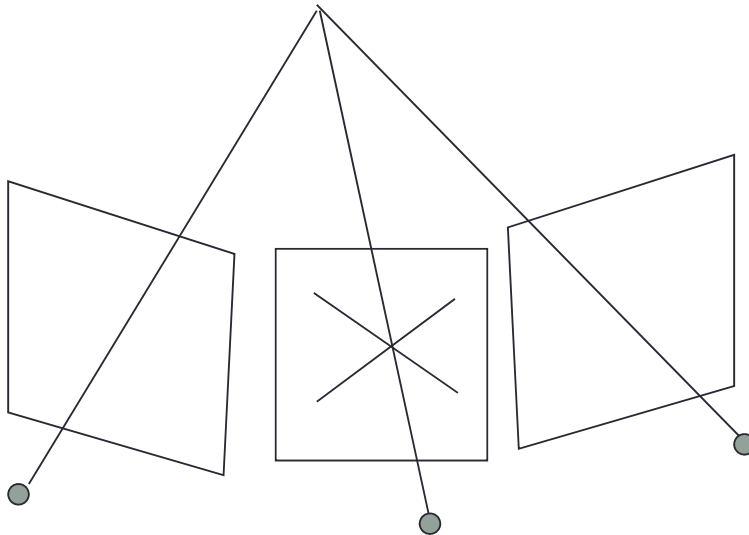
- **κ** : ratio de distancias desde m' a H_m y e'

Paralelaje Proyectivo

- Paralelaje:
 - Directamente proporcional a la distancia al plano
 - Inversamente proporcional a la profundidad
- Datos: F , H
- Cambio de signo (paso por cero) de k indica si el punto está delante o detrás del plano
 - Las transformaciones proyectivas no preservan el orden en profundidad

Homografía entre vistas

- Permite obtener la vista desde otro punto si precisar de un modelo 3D
 - Basta definir una nueva matriz de cámara **P''** y calcular las dos matrices fundamentales (F_{12} , F_{13}) -> matrices de transferencia
 - La nueva imagen se calcula como intersección de las líneas epipolares



$$\tilde{m}'' \cong F_{12}\tilde{m} \wedge F_{23}\tilde{m}'$$

Homografía entre vistas



Video



Reconstrucción Euclídea

- Reconstrucción Directa: Proyectiva $\rightarrow$ Euclídea
 - Si disponemos de al menos 5 puntos de control cuya posición 3D es conocida en el espacio Euclídeo, podemos calcular la transformación proyectiva H_m directamente desde cualquier reconstrucción proyectiva

$$\tilde{m}_i \leftrightarrow \tilde{m}'_i$$

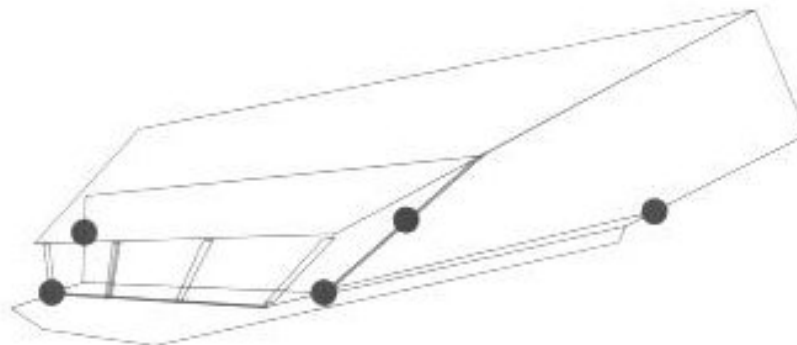
$$(\tilde{P}, \tilde{P}', \{\tilde{M}_i\}) \quad \{\tilde{M}_{Ei}\}$$

$$1) \quad \tilde{M}_{Ei} = H_m \tilde{M}_i$$

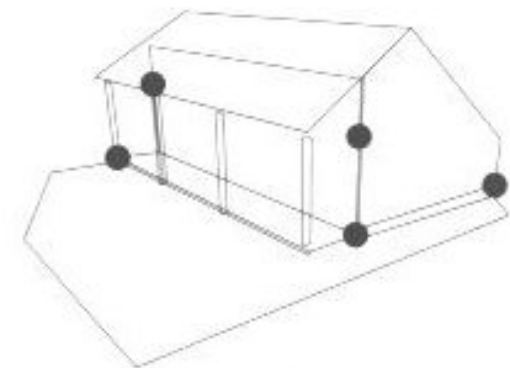
$$2) \quad \begin{aligned} \tilde{m}_{Ei} &= \tilde{P} H_m^{-1} \tilde{M}_{Ei} \\ \tilde{m}'_{Ei} &= \tilde{P}' H_m^{-1} \tilde{M}_{Ei} \end{aligned}$$



a



b



c

Tabla de Contenidos

- Reconstrucción 3D
- Reconstrucción Proyectiva (Autocalibración)
- Odometría Visual

Odometría Visual

- Odometría Visual: localización incremental mediante una cámara (**calibrada**) utilizando múltiples puntos de vista:
 - Robot móvil, dron,
- SLAM: Simultaneous Location and Mapping
 - Incorpora la estimación del mapa
- **Datos:** correspondencias entre puntos proyectados en múltiples planos de imagen: $\tilde{m}_i^{(1)} \leftrightarrow \tilde{m}_i^{(k)}$

$$\tilde{m}_i^{(1)} \cong \tilde{P}_{(1)} \tilde{M}_i \cong A \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} M_i \\ 1 \end{bmatrix} \quad \tilde{m}_i^{(k)} \cong \tilde{P}_{(k)} \tilde{M}_i \cong A \begin{bmatrix} R_{(k)} & t_{(k)} \end{bmatrix} \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

$$\tilde{m}^{(k)T} F \tilde{m}^{(1)} = 0$$

$$\tilde{q}^{(k)T} E \tilde{q}^{(1)} = 0$$

$$E \cong A^T F A \cong [t_{(k)}]_{[\wedge]} \cdot R_{(k)}$$

$R_{(k)} \quad t_{(k)}$ Rotación y traslación (dirección) entre las vistas (1) y (k)

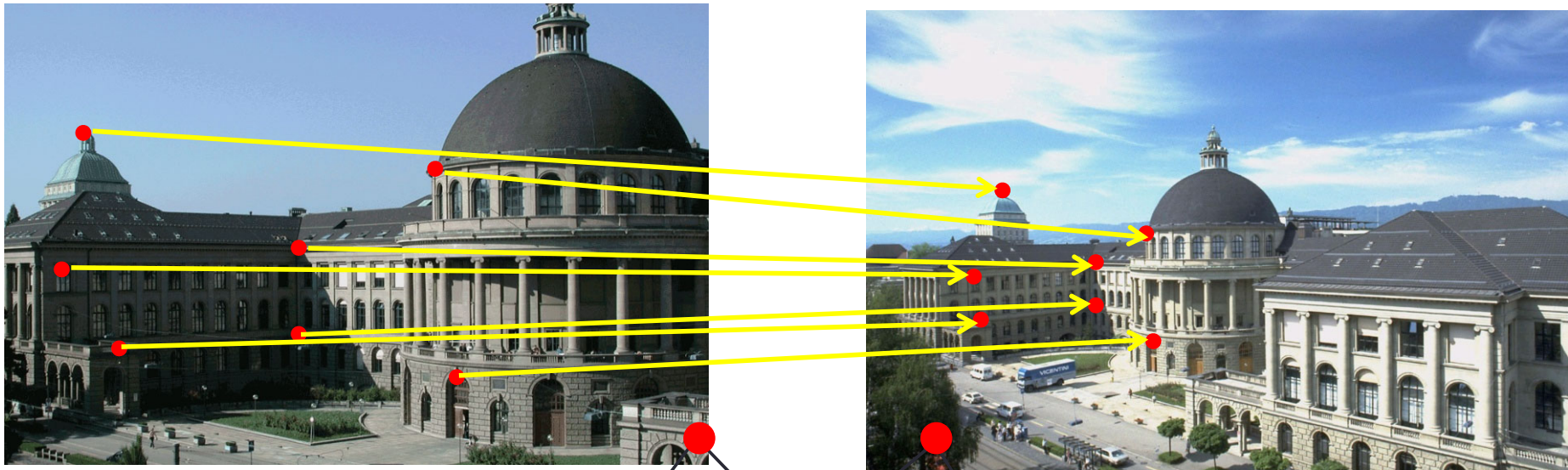
“Visual Odometry: Part I - The First 30 Years and Fundamentals” Davide Scaramuzza

IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011 http://rpg.ifi.uzh.ch/docs/VO_Part_I_Scaramuzza.pdf

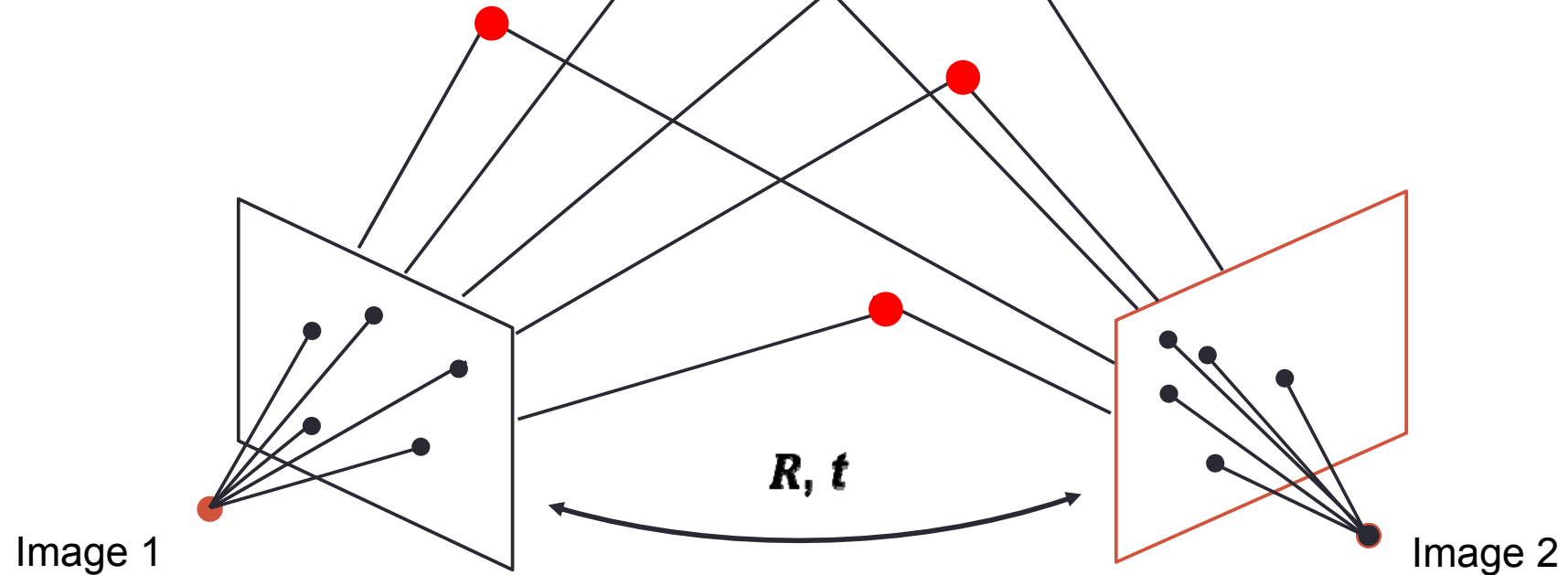
“Visual odometry: Part II - Matching, robustness, optimization, and applications” Davide Scaramuzza

IEEE Robotics and Automation Magazine, Volume 19, issue 2, 2012 http://rpg.ifi.uzh.ch/docs/VO_Part_II_Scaramuzza.pdf

Odometría Visual



(D. Scaramuzza Tutorial)



Odometría Visual

- Odometría Visual: localización incremental mediante una cámara (**calibrada**) utilizando múltiples puntos de vista:
 - Robot móvil, dron,



(Dyson Robotics 360, Andrew Davison, Imperial College, London)

Video

Odometría Visual

- Odometría Visual: localización incremental mediante una cámara (**calibrada**) utilizando múltiples puntos de vista:

- Cálculo incremental:

$$\tilde{m}_i^{(k)} \leftrightarrow \tilde{m}_i^{(k+1)}$$

$$\tilde{m}_i^{(k)} \cong A \begin{bmatrix} R_{(k)} & t_{(k)} \end{bmatrix} \begin{bmatrix} M_i \\ 1 \end{bmatrix} \quad \tilde{m}_i^{(k+1)} \cong A \begin{bmatrix} R_{(k+1)} & t_{(k+1)} \end{bmatrix} \begin{bmatrix} M_i \\ 1 \end{bmatrix}$$

${}^{(k)}R_{(k+1)} \quad {}^{(k)}t_{(k+1)}$ Rotación y traslación (dir.) entre las vistas (k) y (k + 1)

$$\tilde{m}^{(k+1)T} F \tilde{m}^{(k)} = 0$$

$$E \cong A^T F A \cong \begin{bmatrix} {}^{(k)}t_{(k+1)} \end{bmatrix}_{[\wedge]} \cdot {}^{(k)}R_{(k+1)}$$

- Actualización de la odometría:

$$R_{(k+1)} = R_{(k)} \cdot {}^{(k)}R_{(k+1)}$$

$$t_{(k+1)} = t_{(k)} + R_{(k)} \cdot {}^{(k)}t_{(k+1)}$$

- La traslación (escala) se calcula a partir de la distancia entre los puntos 3D reconstruidos (mapa) :

$$\{M_i\}^k \leftrightarrow \{M_i\}^{k+1} \Rightarrow {}^{(k)}t_{(k+1)}$$

Odometría Visual: Esquema general

- Algoritmo:

- Calcular la matriz Fundamental/Esencial a partir de la correspondencia de puntos (Calibración Proyectiva) (Si nos movemos en un plano se puede utilizar el algoritmo de 3 puntos):

$$\begin{aligned} \tilde{m}^{(k+1)T} F \tilde{m}^{(k)} &= 0 & E &= A^T F A \\ \tilde{m}_i^{(k)} \leftrightarrow \tilde{m}_i^{(k+1)} & & \tilde{q}^{(k+1)T} E \tilde{q}^{(k)} &= 0 & \tilde{q} &= A^{-1} \tilde{m} \end{aligned}$$

- Descomposición de la matriz esencial en una matriz anti-simétrica y una matriz de rotación (4 posibilidades):

$$E \cong [t]_{[\wedge]} \cdot R$$

- Reconstrucción 3D de los puntos (mapa) para las cuatro posibilidades: $\{ M_i \}$

- Nos quedamos con aquella en la que los puntos 3D están por delante de la cámara ($Z > 0$). Rotación y dirección de traslación:

$${}^{(k)}R_{(k+1)} ; \quad \lambda \cdot {}^{(k)}t_{(k+1)}$$

- Actualización de la odometría:

$$R_{(k+1)} = R_{(k)} \cdot {}^{(k)}R_{(k+1)}$$

- La traslación se calcula a partir de los puntos 3D reconstruidos (PnP)
 - Se calcula entre dos posiciones en las que la traslación (línea base) es grande para reducir el error de estimación

$$t_{(k+i)} = t_{(k)} + R_{(k)} \cdot {}^{(k)}t_{(k+i)}$$

Odometría Visual

- Incertidumbre proyectiva (factor de escala)
 - A partir de dos vistas no podemos extraer la distancia de desplazamiento, solo la dirección
- El mapa y la localización están definidas salvo un factor de escala (Trasformación de Similitud)
 - Una escena con un objeto grande a gran distancia y un desplazamiento de cámara grande, genera los mismos puntos proyectados que una escena con un objeto pequeño cercano y un desplazamiento de cámara pequeño
- Para sucesivas vistas podemos calcular el factor de escala en el siguiente paso, como el cociente de distancias entre puntos del mapa 3D generado respecto al anterior.
- El factor de escala inicial se estima:
 - A partir de información métrica de la imagen: un marcador de tamaño conocido
 - Un sensor adicional: IMU, acelerómetro

Odometría Visual

- Incertidumbre proyectiva (factor de escala)

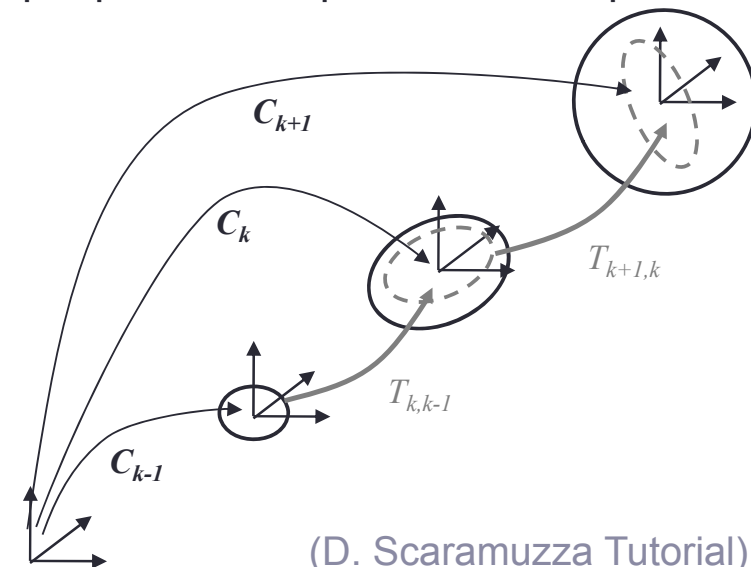


(Habitación de Ames)

Video

Odometría Visual

- Problemas de implementación:
 - Para que la reconstrucción 3D (mapa) sea robusta es preciso que la distancia entre puntos de vista sea elevada, pero esto dificulta la correspondencia
 - SLAM: se utilizan Key-frames:
 - Se realiza el tracking y el cálculo odométrico para cada imagen capturada
 - El mapa solo se actualiza cuando la distancia entre puntos de vista es suficiente elevada
- Propagación del Error:
 - El error de estimación se va acumulando lo que provocará que con el tiempo la odometría sea incorrecta
 - Filtro de Kalman (EKF), Filtro de Partículas

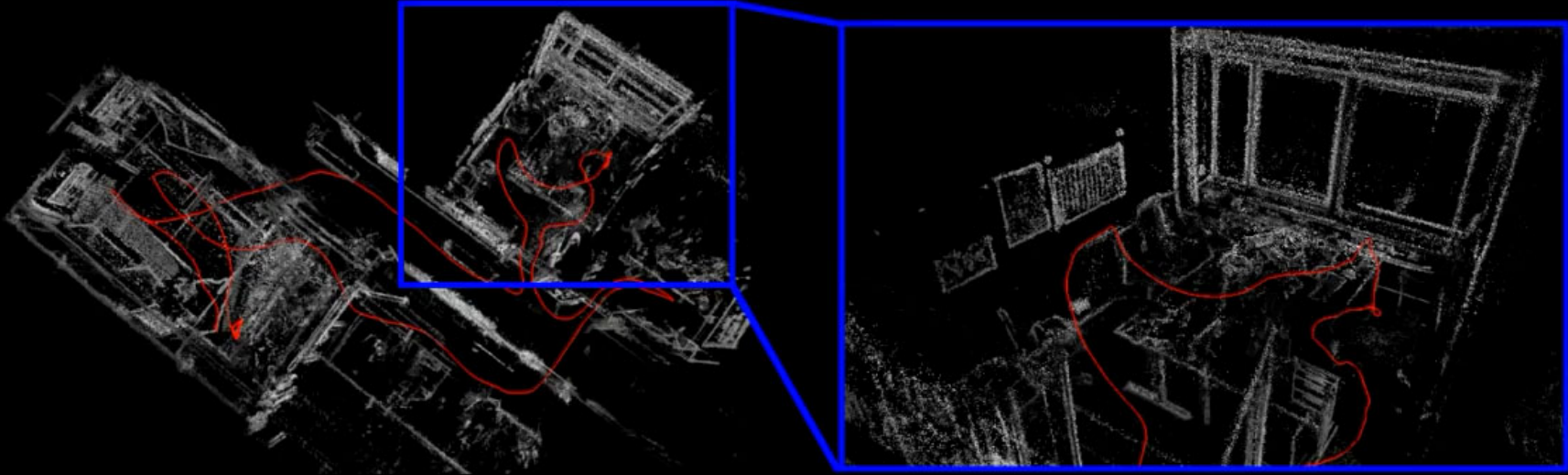


(D. Scaramuzza Tutorial)

Odometría Visual

- Ejemplos:

Direct Sparse Odometry
Jakob Engel^{1,2}, Vladlen Koltun², Daniel Cremers¹
July 2016



TUM¹Computer Vision Group
Technical University Munich

²Intel Labs 

Tabla de Contenidos

- Reconstrucción 3D
- Reconstrucción Proyectiva (Autocalibración)
- Odometría Visual