# Robótica Móvil

## Visión por computadora

Taihú Pire

Laboratorio de Robótica

CONICET

UNIVERSIDAD NACIONAL DE ROSARIO
CONFINGERE HOMINEM COGITANTEM

CIFASIS

# Motivación

Las cámaras pueden ser utilizadas para realizar una reconstrucción del entorno y estimar la localización de un robot. Visual Odometry / Visual SLAM.

**Agregar imagen o video de Visual SLAM o de Visual Odometry**

# Detección de Keypoints

Propiedades deseables para SLAM / SfM:

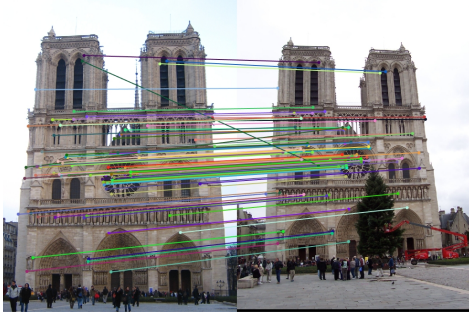
- Alta repetitibilidad
- Invarianza (luminosidad, puntos de vista, etc)
- Computacionalmente eficientes
- Algunos detectores: HARRIS, SIFT, SURF, Shi-Tomasi, FAST, ORB, BRISK

Keypoints FAST

# Matcheo de Keypoints

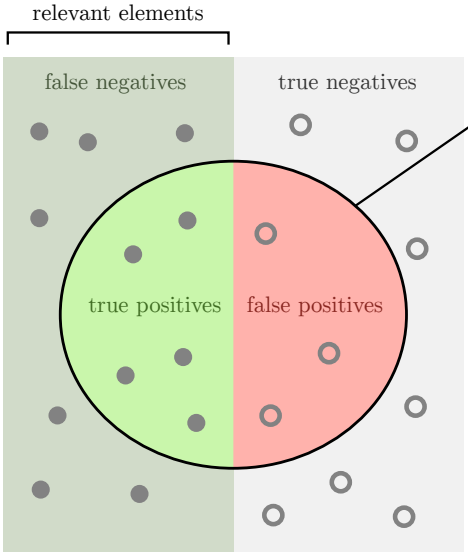
Propiedades deseables del matcheo de keypoints para SLAM / SfM:

- Alto recall
- Precisión
- Robustez
- Computacionalmente eficientes
- Posibles enfoques: patches o descriptores



Keypoints Harris + Descriptor SIFT

# Precision - Recall




How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{retrieved elements}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

# Descriptores de Features Locales

Propiedades deseables para SLAM / SfM: distinguibles, robustos, invariantes

- Extraer firmas sobre regiones de la imagen, ejemplos:
  - Histogramas sobre gradientes de la imagen (SIFT)
  - Histogramas sobre respuestas Haar-wavelet (SURF)
  - Patrones binarios (BRIEF, BRISK, FREAK, ORB)
  - Descriptores basados en aprendizaje
- Invariante rotación: Alineado con la orientación dominante de la región local
- Invariante a escala: Adaptar la región descripta a la escala de keypoint

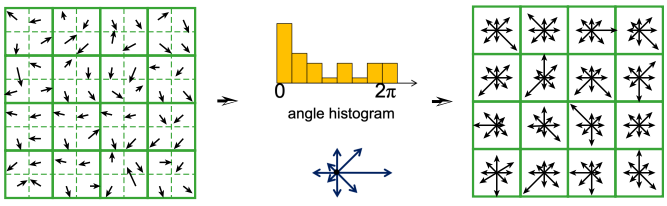
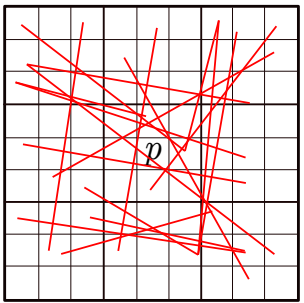



image gradients

keypoint descriptor

Patrón BRIEF en Patch de 9x9



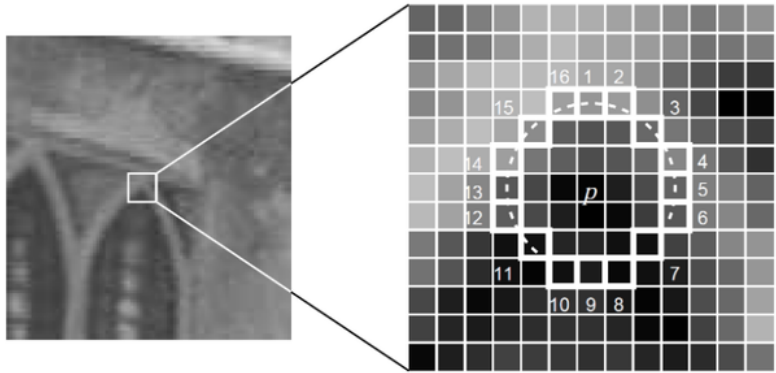
# Invarianza de Features Locales

- Invarianza geométrica: traslación, rotación, escala.
- Invarianza fotométrica: brillo, exposición, etc.

# Ventajas de features locales

- **Locales**: los features al ser locales son robustos a oclusiones
- **Distintivos**: pueden diferenciar un gran conjunto de objetos
- **Cuantiosos**: puede haber cientos o miles en una misma imagen
- **Eficientes**: al discretizar la imagen, se puede obtener tiempo real.

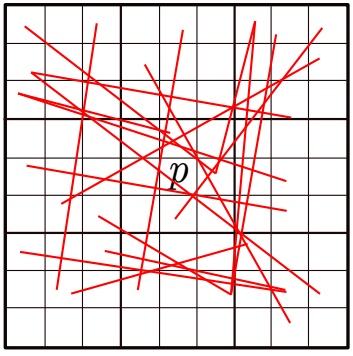
# Características Visuales: Detector FAST




- El valor de intensidad del pixel $p$ es comparado con cada uno de los 16 pixeles del círculo de Bresenham alrededor de $p$.
- $p$ es detectado como corner si hay $12$ píxeles continuos en el circulo de Bresenham más brillosos u oscuros que $p$ dado un cierto umbral.

# Características Visuales: Descriptor BRIEF

Patrón BRIEF en Patch de 9x9




256 Comparaciones entre píxeles (1 a 1)

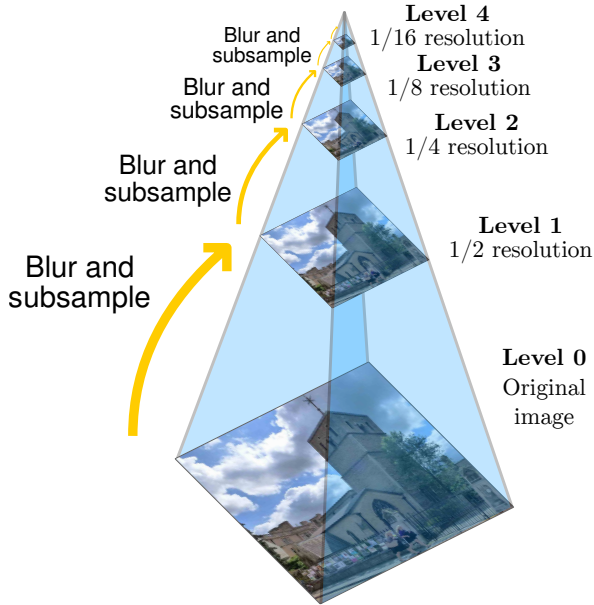
$$\tau(p; x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases}$$

$$s = \overbrace{01010010101110010...}^{256 \text{ bits}}$$

## Matching distance

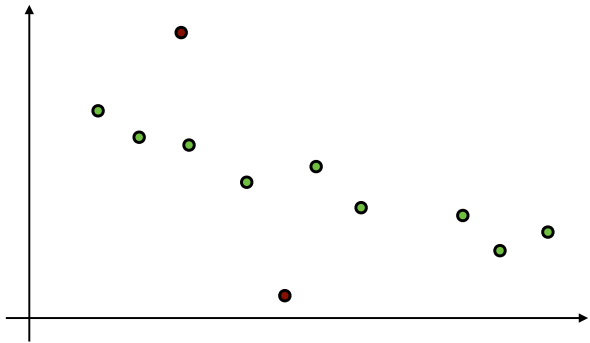
Hamming distance $= sum(XOR(s_1, s_2))$

# Extracción de features a diferentes escalas



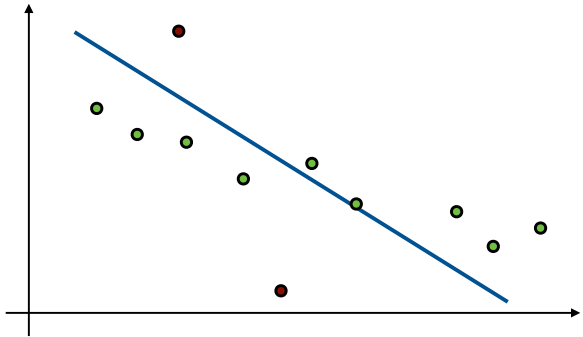
# RANSAC: Random Sample Consensus

- Permite encontrar un modelo que ajusta datos en presencia de ruido y outliers
- Separa el conjunto de datos entre inliers y outliers
- Ejemplo: dado un conjunto de puntos 2D, ajustar una línea



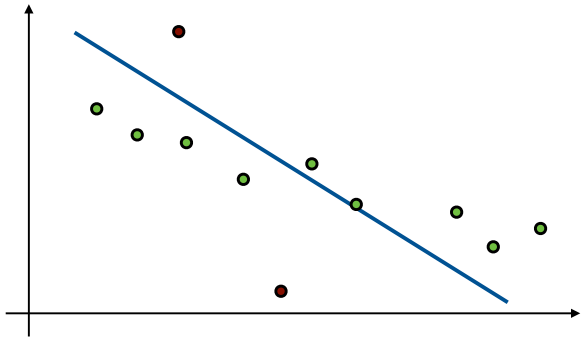
# RANSAC: Random Sample Consensus

Podemos Ajustar una recta utilizando mínimos cuadrados, asumiendo ruido constante para todos los puntos



# RANSAC: Random Sample Consensus

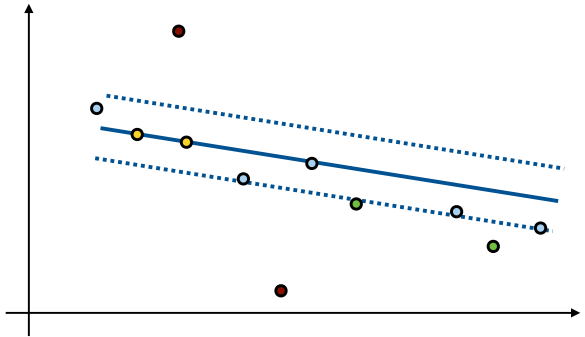
Podemos Ajustar una recta utilizando mínimos cuadrados, asumiendo ruido constante para todos los puntos



Solo necesitamos 2 puntos para ajustar una recta! Probemos con 2 puntos aleatorios...

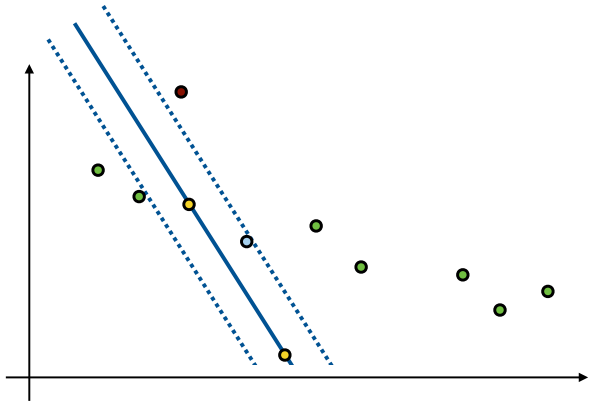
# RANSAC: Random Sample Consensus

Probando con 2 puntos aleatorios



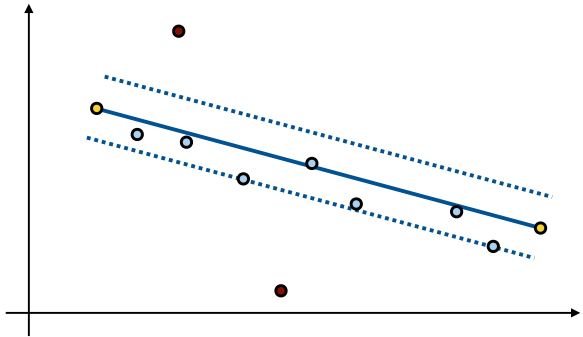
# RANSAC: Random Sample Consensus

Probando con otros 2 puntos aleatorios



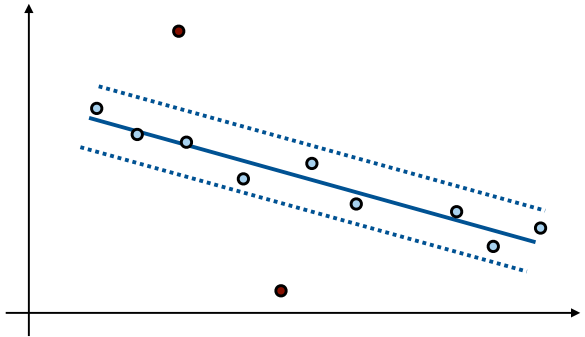
# RANSAC: Random Sample Consensus

Probando con otros 2 puntos aleatorios



# RANSAC: Random Sample Consensus

Utilicemos los inliers obtenidos con el mejor intento hasta ahora para realizar un ajuste con mínimos cuadrados



# RANSAC: Random Sample Consensus

1. **Samplear** de manera aleatoria el número de puntos requerido para ajustar el modelo.
2. **Computar** el modelo usando los datos sampleados
3. **Contar** el número de inliers y quedarse con el modelo que mejor ajusta los datos
4. Iterar pasos 1-3 hasta que el mejor modelo es hallado

# RANSAC: Random Sample Consensus

Pero...¿cuantas iteraciones realizar?

- Número de puntos sampleados $s$ (número de puntos mínimos requeridos para ajustar el modelo)
- Ratio de outliers $e = \frac{\#outliers}{\#puntostotales}$
- Número de intentos $T$. Elegimos $T$, con probabilidad $p$ de éxito (la probabilidad de al menos obtener un muestreo aleatorio libre de outliers en las $T$ iteraciones).

Probabilidad de fallar en una iteración, es decir de no seleccionar todos inliers.

$$1 - p = 1 - (1 - e)^s$$

Probabilidad de fallar en $T$ iteraciones, es decir seleccionar un outlier en todas las iteraciones.

$$1 - p = (1 - (1 - e)^s)^T$$

despejando $T$,

$$T = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)}$$

# RANSAC: ventajas y desventajas

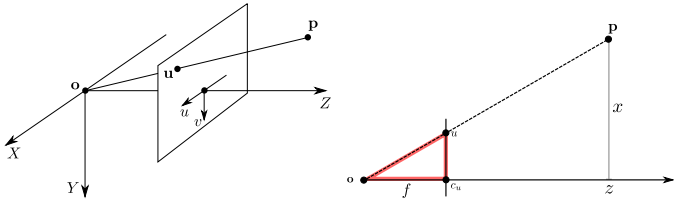
Ventajas:

- Robusto en presencia de outliers
- funciona bien para modelos de 1 a 10 parámetros (dependiendo del número de outliers)
- Fácil de implementar y entender

Desventajas:

- El tiempo computacional crece rápido con el porcentaje de outliers y el número de parámetros necesarios para ajustar el modelo
- No es bueno para obtener múltiples modelos (e.g. ajustar más de una línea en 2D)

# Cámara - Modelo Pin-Hole

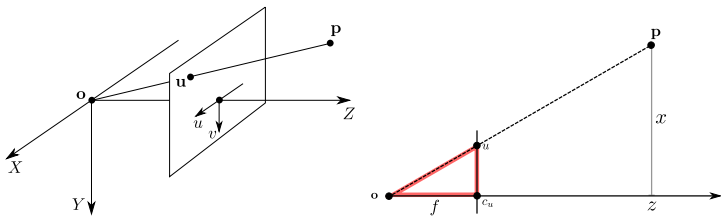



## Modelo Pin-hole básico

Una Cámara permite obtener correspondencias entre el mundo 3D (espacio del objeto) y la imagen 2D. En el modelo de cámara pinhole, el punto de la imagen $\mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^\top$ se determina como la intersección entre el **plano de la imagen** y el rayo que une el punto del mundo $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ y el **centro de proyección óptico** $\mathbf{o}$.

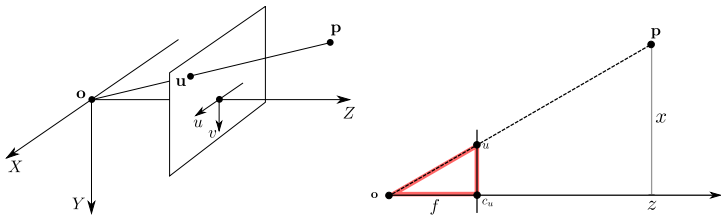
- $f$ es la **distancia focal**
- $\mathbf{c} = \begin{bmatrix} 0 & 0 & f \end{bmatrix}^\top$ es el **punto principal**
- el eje $Z$ es el **eje principal**
- ${}^{\mathrm{W}}\mathbf{p}$ es el punto en el sistema de coordenadas del mundo

# Cámara - Modelo Pin-Hole




- El plano $Z = f$ se denomina **plano principal** o **plano de la imagen**.
- El punto en la imagen $\mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^\top$, en coordenadas 3D es $\mathbf{u} = \begin{bmatrix} x & y & f \end{bmatrix}^\top$

# Cámara - Modelo Pin-Hole




Para obtener las coordenadas $\mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^\top$ del punto ${}^{\mathrm{W}}\mathbf{p}$ en el plano de la imagen, se proyecta ${}^{\mathrm{W}}\mathbf{p}$ sobre el plano $ZX$ y $ZY$. Por similitud de triángulos para cada proyección:

$$\frac{x}{z} = \frac{u}{f} \quad \text{y} \quad \frac{y}{z} = \frac{v}{f}$$

Por tanto, tenemos:

$$u = \frac{fx}{z} \quad \text{y} \quad v = \frac{fy}{z}$$

# Cámara - Modelo Pin-Hole

La correspondencia entre un punto $\mathbb{R}^3$ y $\mathbb{R}^2$ está dada por

$$\begin{bmatrix} x & y & z \end{bmatrix}^\top \rightarrow \begin{bmatrix} u = \frac{fx}{z} & v = \frac{fy}{z} \end{bmatrix}^\top$$

Está correspondencia está dada por una transformación $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ tal que

$$\mathbf{P}\, {}^{\mathrm{W}}\dot{\mathbf{p}} = \dot{\mathbf{u}}$$

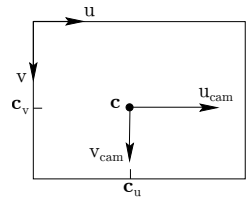
donde $\mathbf{P}$ es la matríz asociada y se la denomina **matríz de proyección**.

$$\begin{bmatrix} x & y & z \end{bmatrix}^\top \rightarrow \begin{bmatrix} u = \frac{fx}{z} & v = \frac{fy}{z} \end{bmatrix}^\top \tag{1}$$

Sea ${}^{\mathrm{W}}\dot{\mathbf{p}} \in \mathbb{P}^3$, ${}^{\mathrm{W}}\dot{\mathbf{p}} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top$ entonces

$$\begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

# Cámara - Modelo Pin-Hole




Sin embargo si el origen de coordenadas del plano de la imagen no coincide con el punto principal $\mathbf{c} = \begin{bmatrix} 0 & 0 & f \end{bmatrix}^\top$, se realiza una traslación y la correspondencia es:

$$\begin{bmatrix} x & y & z \end{bmatrix}^\top \rightarrow \begin{bmatrix} u = \frac{fx}{z} + \mathbf{c}_u & v = \frac{fy}{z} + \mathbf{c}_v \end{bmatrix}^\top \tag{3}$$

Utilizando matrices

$$\begin{bmatrix} fx + z\mathbf{c}_u \\ fy + z\mathbf{c}_v \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & \mathbf{c}_u & 0 \\ 0 & f & \mathbf{c}_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4}$$

# Cámara - Modelo Pin-Hole

$$\begin{bmatrix} fx + z\mathbf{c}_u \\ fy + z\mathbf{c}_v \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & \mathbf{c}_u & 0 \\ 0 & f & \mathbf{c}_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} fx + z\mathbf{c}_u \\ fy + z\mathbf{c}_v \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & \mathbf{c}_u \\ 0 & f & \mathbf{c}_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

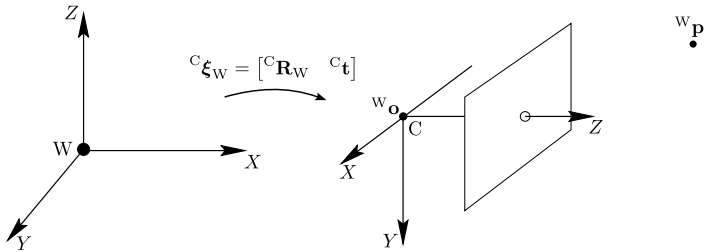
Usando notación matricial, nos queda

$$\underset{3\times4}{\mathbf{P}} = \underset{3\times3}{\mathbf{K}} \underset{3\times4}{\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} \end{bmatrix}},$$

donde $\mathbf{K}$ es la **matriz de calibración** o **matriz intrínseca** de la cámara y está definida como,

$$\mathbf{K} = \begin{bmatrix} f & 0 & \mathbf{c}_u \\ 0 & f & \mathbf{c}_v \\ 0 & 0 & 1 \end{bmatrix}$$

# Cámara - Modelo Pin-Hole




En el caso general, la cámara no está ubicada en el centro del sistema de coordenadas del mundo. Tenemos que agregar esta transformación (rotación y traslación) entre el mundo y los sistemas de coordenadas de la cámara a la transformación.

Dado el punto $^{W}\mathbf{p}$ en coordenadas del mundo $W$ , el mismo punto está descripto en el sistema de coordenadas de la cámara $C$, y se lo representa $^{C}\mathbf{p}$. Esto significa que $^{C}\mathbf{p}$ está escrito en un sistema cuyo origen es $^{W}\mathbf{o}$ y los ejes de de coordenadas sufrieron una rotación $^{C}\mathbf{R}_{W}$:

$$^{C}\mathbf{p} = {}^{C}\mathbf{R}_{W}\left({}^{W}\mathbf{p} - {}^{W}\mathbf{o}\right)$$

# Cámara - Modelo Pin-Hole

En coordenadsa homogeneas:

$$ {}^{\mathrm{C}}\dot{\mathbf{p}} = {}^{\mathrm{C}}\begin{bmatrix} {}^{\mathrm{C}}\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} ({}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}}\,{}^{\mathrm{W}}\mathbf{p} - {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}}\,{}^{\mathrm{W}}\mathbf{o}) \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}} & -{}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}}\,{}^{\mathrm{W}}\mathbf{o} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{W}}\mathbf{p} \\ 1 \end{bmatrix} $$

Reescribiendo ${}^{\mathrm{C}}\mathbf{t} = -{}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}}\,{}^{\mathrm{W}}\mathbf{o}$

$$ {}^{\mathrm{C}}\dot{\mathbf{p}} = {}^{\mathrm{C}}\begin{bmatrix} {}^{\mathrm{C}}\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}} & {}^{\mathrm{C}}\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^{\mathrm{W}}\mathbf{p} \\ 1 \end{bmatrix} $$

Luego si, $\dot{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} {}^{\mathrm{C}}\dot{\mathbf{p}}$ y ${}^{\mathrm{C}}\dot{\mathbf{p}} = \begin{bmatrix} {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}} & {}^{\mathrm{C}}\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} {}^{\mathrm{W}}\dot{\mathbf{p}}$, entonces

$$ \dot{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \begin{bmatrix} {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}} & {}^{\mathrm{C}}\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} {}^{\mathrm{W}}\dot{\mathbf{p}} $$

La matriz de proyección para el caso en que la cámara pueda ubicarse fuera del origen del mundo entonces queda:

$$ \mathbf{P} = \mathbf{K}\begin{bmatrix} {}^{\mathrm{C}}\mathbf{R}_{\mathrm{W}} & {}^{\mathrm{C}}\mathbf{t} \end{bmatrix} $$

# Cámara CCD

En cámaras CCD, los pixels pueden no ser cuadrados. Si las coordenadas de la imagen son medidas en píxeles, entonces hay que tener en cuenta las dimensiones del pixel en cada dirección.
Sean

- $m_u$ : número de píxeles por unidad de distancia en coordenadas de la imagen en la dirección $u$
- $m_v$ : el número de píxeles por unidad de distancia en coordenadas de la imagen en la dirección $v$

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & \mathbf{c}_u \\ 0 & f_v & \mathbf{c}_v \\ 0 & 0 & 1 \end{bmatrix},$$

donde $f_u = fm_u$ y $f_v = fm_v$ representan la distancia focal de la cámara en términos de dimensiones de píxeles en la dirección $u$ e $v$ respectivamente.

# Cámara proyectiva finita

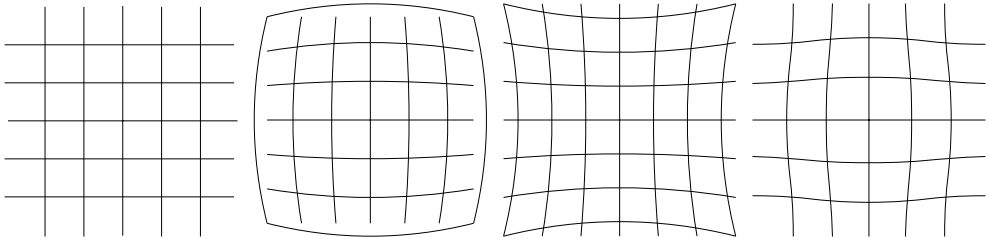
Agregando más generalidad, puede ser que los píxeles estén "torcidos", aunque no es el caso usual.


Para esto se agrega el parámetro *skew* $s$ a la matriz de calibración

$$\mathbf{K} = \begin{bmatrix} f_u & s & \mathbf{c}_u \\ 0 & f_v & \mathbf{c}_v \\ 0 & 0 & 1 \end{bmatrix},$$

# Distorción Radial

Previamente, consideramos que el modelo de cámara pin-hole es perfectamente lineal, es decir, el punto en el mundo, el punto en la imagen y el centro óptico son colineales.
Esta suposición no es cierta en la práctica porque hay una distorsión radial en la imagen producida por la lente de la cámara real.



Sin distorción | Barrel distortion | Pincushion distortion | Mustache distortion

# Distorción Radial

Para corregir esta distorsión se calcula un modelo de distorsión radial que relaciona los puntos de la imagen de la imagen real con los puntos ideales, los que se habrían obtenido bajo una cámara lineal perfecta. De esta forma, la cámara se puede representar con un modelo pin-hole lineal. En la figura es posible observar la imagen cruda (distorsionada) tomada por la cámara y la imagen después de la desdistorsión.

Imagen distorcionada

Imagen desdistorcionada

# Distorción Radial

Dado el punto proyectado real y el punto ideal, el efecto de distorsión radial se puede modelar mediante

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = L(\tilde{r}) \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} \tag{5}$$

donde, $\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix}$ es la posición ideal de la imagen referida al centro de la imagen (que obedece a la proyección lineal); $\begin{bmatrix} u_d \\ v_d \end{bmatrix}$ es la posición real de la imagen después de la distorsión radial; $\tilde{r}$ es la distancia radial $\sqrt{\tilde{u}^2 + \tilde{v}^2}$ desde el centro para la distorsión radial, y $L(\tilde{r})$ es una factor de distorsión, que es una función del radio $\tilde{r}$.

# Distorción Radial

**Agregar modelo de distorción aplicado en: https://boofcv.org/index.php?title=Tutorial_Camera_Calibration**

Finalmente, la corrección se puede representar en coordenadas de píxeles como

$$\hat{u} = u_c + L(r)(u - u_c)$$
$$\hat{v} = v_c + L(r)(v - v_c)$$

donde $\begin{bmatrix} u \\ v \end{bmatrix}$ son las coordenadas medidas, $\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix}$ son las coordenadas corregidas, y $\begin{bmatrix} u_c \\ v_c \end{bmatrix}$ es el centro de distorsión radial, con $r^2 = (u - u_c)^2 + (v - v_c)^2$.

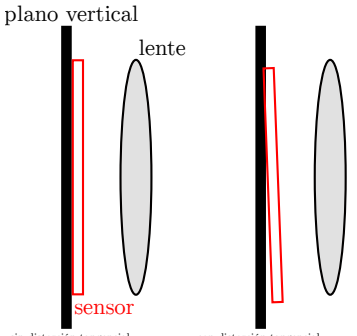
El factor de distorsión $L(r)$ esta definido solo por valores positivos de $r$ y $L(0) = 1$. En la práctica, $L(r)$ es aproximado por la expansión de Taylor

$$L(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots.$$

Los coeficientes $\{k_1, k_1, k_1, \dots, u_c, v_c\}$ se consideran parte de la calibración interna de la cámara. Con frecuencia, el punto principal se utiliza como centro de la distorsión radial, aunque no es necesario que coincidan exactamente. Los coeficientes de corrección, junto con la matriz de calibración de la cámara, especifican el mapeo desde un punto de la imagen hasta un rayo en el sistema de coordenadas de la cámara y se denominan parámetros intrínsecos.

# Distorción Tangencial

**Agregar matemática de distorción tangencial**



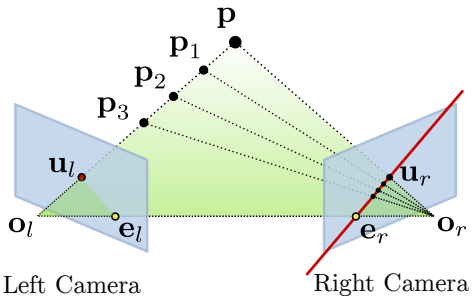
# Geometría Epipolar

- La restricción epipolar puede ser computada por medio de puntos 2D en la imagen cuando la pose relative de la cámara es conocida ($[\mathbf{t}]_\times \mathbf{R}$, por ejemplo una cámara estéreo)
- El **plano epipolar** es definido por $\mathbf{u}_l$ y los dos centros óptico de las cámaras $\mathbf{o}_l$ y $\mathbf{o}_r$
- La **línea epipolar** es la intersección del plano epipolar y el plano de la imagen derecha.
- La restricción epipolar codifica que $\mathbf{u}_r$ debe estar ubicado sobre la linea epipolar en la imagen derecha.

$$\dot{\mathbf{u}}_l^\top \mathbf{E} \dot{\mathbf{u}}_r = 0 \quad \text{(restricción de coplanaridad)}$$

donde $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$, y se denomina *Essential Matrix*

La línea epipolar permite que restrinjamos la búsqueda de correspondencias visuales entre dos cámaras.



# Matríz fundamental

La **Matríz Esencial** es una especialización de la **Matríz Fundamental**. La Matríz Esencial trabaja sobre **coordenadas normalizadas** de la imagen, es decir, requiere tener la calibración de las cámaras, mientras que la Matríz Fundamental relaciona correspondencias desconociendo las calibraciones.

$$\dot{\mathbf{u}}'^{\top}\mathbf{F}\dot{\mathbf{u}} = 0 \quad \text{(restricción de coplanaridad)}$$

## Coordenadas Normalizadas:

Dado $\dot{\mathbf{u}} = \mathbf{P}\,{}^{\mathrm{W}}\dot{\mathbf{p}}$, con $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, si conocemos la matríz de calibración $\mathbf{K}$ entonces $\hat{\mathbf{u}} = \mathbf{K}^{-1}\dot{\mathbf{u}}$ decimos que $\hat{\mathbf{u}}$ está en **coordenadas normalizadas**. Se lo puede pensar como una imagen del punto ${}^{\mathrm{W}}\dot{\mathbf{p}}$ con respecto a una cámara $[\mathbf{R}|\mathbf{t}]$ teniendo la matriz de calibración como la identidad. $\mathbf{K}^{-1}\mathbf{P} = [\mathbf{R}|\mathbf{t}]$ se denomina **cámara normalizada** donde los parámetros de calibración han sido removidos.

**corregir cuentas.**

$$\dot{\mathbf{u}}'^{\top}\mathbf{E}\dot{\mathbf{u}} = 0$$
$$\left(\mathbf{K}'^{-1}\dot{\mathbf{u}}'\right)^{\top}\mathbf{E}\left(\mathbf{K}^{-1}\dot{\mathbf{u}}\right) = 0$$
$$\dot{\mathbf{u}}'^{\top}\mathbf{K}'^{\top-1}\mathbf{E}\mathbf{K}^{-1}\dot{\mathbf{u}} = 0$$

La relación entre la matriz fundamental y la matríz esencial está dada por

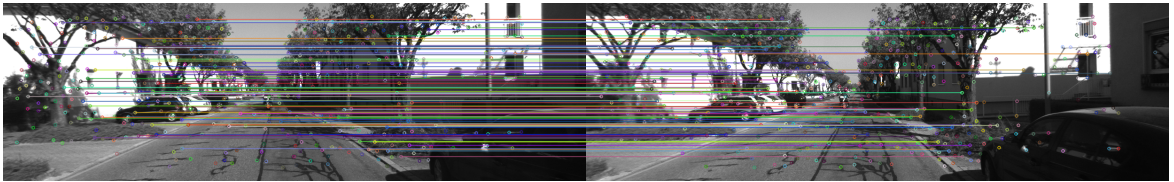
$$\mathbf{E} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K}$$

# Matríz Esencial

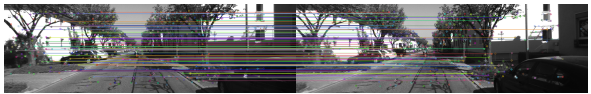
**Agregar imagen de las posibles disposiciones que da la matriz esencial entre dos cámaras. En total da 4 disposiciones posibles para las 2 cámaras. Nos quedamos con la disposición que haga que los puntos triangulados esten delante de la cámara.**

# Matríz Fundamental vs Matríz Esencial

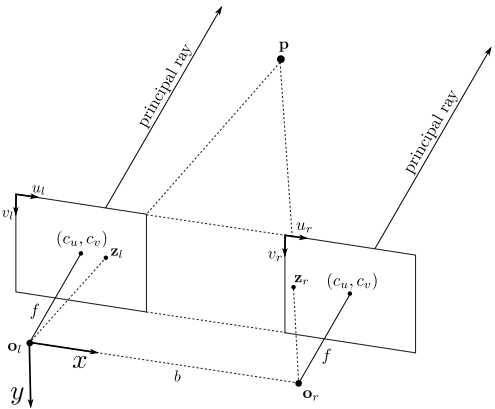
- Ambas matrizes tienen rango 2
- La matríz esencial tiene 5 DoF
- La matríz fundamental tiene 7 DoF (ya que tience información sobre los parámetros de calibración)
- Las restricciones coplanares más pares de correspondencias nos permiten encontrar dichas matrices por medio de la solución de un sistema líneal utilizando descomposición SVD (*Singular Value Decomposition*).
- La matríz fundamental puede ser hallada con *8-Point Algorithm*
- La matríz esencial puede ser hallada con *5-Point Algorithm*
- Una vez obtenida la matríz esencial podemos obtener la rotación y traslación (con ambigüedad de escala)
- La matríz esencial o fundamental nos permiten delimitar la búsqueda de correspondencias en las líneas epipolares únicamente, en vez de tener que buscar en toda la imagen.



# Triangulación estéreo (con cámara rectificada)



Matching estéreo.




Triangulación estéreo.

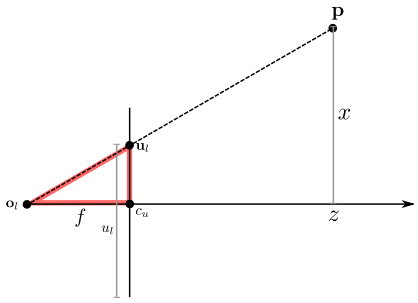
$$\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$$

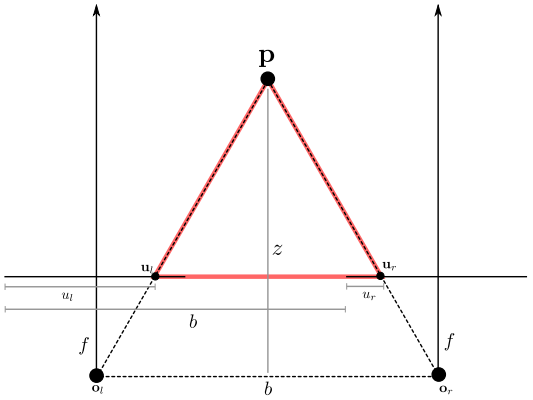
$$x = \frac{(u_l - c_u)z}{f}$$

$$y = \frac{(v_l - c_v)z}{f}$$

$$z = \frac{bf}{u_l - u_r}$$

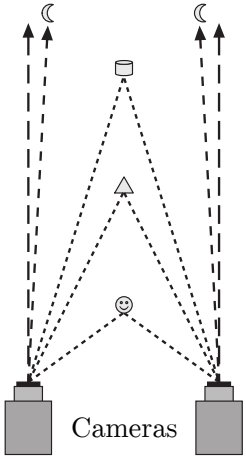
# Triangulación estéreo (con cámara rectificada)

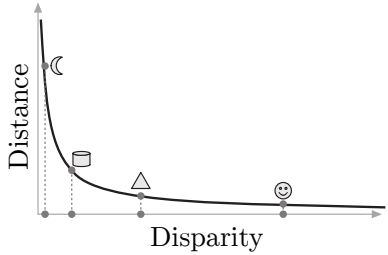



$$\frac{b}{z} = \frac{b - (\mathbf{u}_l - \mathbf{u}_r)}{z - f} \Rightarrow z = \frac{fb}{(\mathbf{u}_l - \mathbf{u}_r)} = \frac{fb}{d}$$

# Disparidad

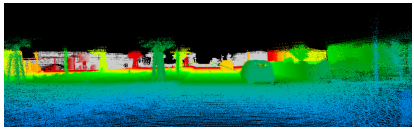


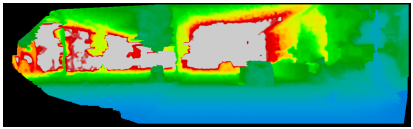

Ejemplo de disparidad: pongamos un dedo enfrente de nuestros ojos a una distancia $20\,\text{cm}$ y alternadamente cerremos y abramos cada ojo.

# KITTI: error de reconstrucción

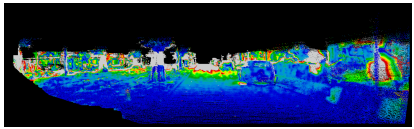
Left image



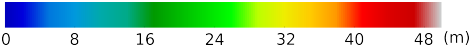
Ground-truth depth



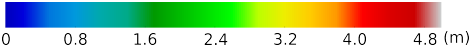
LIBELAS depth



LIBELAS depth errors




Color metric for depth maps. Gray stands for the largest values.



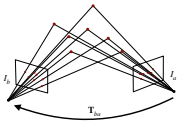

Color metric for depth errors. Gray stands for the largest values.

Comparison of LIBELAS depth maps against the ground truth, for a single frame (KITTI dataset).

# Estimación de movimiento

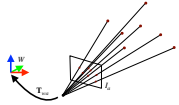
**2D-2D**

- Error de reproyección:
  $f({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}, P) = \sum_{i=1}^{N} \|\mathbf{z}_{a,i} - \hat{\mathbf{z}}^{\mathrm{s}}({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{W}}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2 + \|\mathbf{z}_{b,i} - \hat{\mathbf{z}}^{\mathrm{s}}({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}{}^{-1}\,{}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{W}}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2$
- Algoritmos lineales: 8-point, 5-point



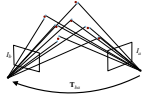
**3D-2D**

- Error de reproyección:
  $$f({}^{\mathrm{W}}\boldsymbol{\xi}_{\mathrm{a}}) = \sum_{i=1}^{N} \|\mathbf{z}_{a,i} - \hat{\mathbf{z}}({}^{\mathrm{W}}\boldsymbol{\xi}_{\mathrm{a}}{}^{-1}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2$$
- Algoritmos lineales: DLT, PnP



**3D-3D**

- Error de reproyección:
  $$f({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}) = \sum_{i=1}^{N} \| {}^{\mathrm{a}}\mathbf{p}_i - {}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{p}_i\|^2$$
- Algoritmos lineales: Atun, Horn



# Estimación de movimiento 2D-2D

- Dados matches 2D-2D $\{(\mathbf{z}_a, \mathbf{z}_b)_i\}$ de puntos 3D desconocidos $P_i$ encontrar el movimiento relativo ${}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}$ entre los frames.
- Error de reproyección (Bundle Adjustment):

$$f({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}, P) = \sum_{i=1}^{N} \|\mathbf{z}_{a,i} - \hat{\mathbf{z}}^{\mathrm{s}}({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{W}}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2 + \|\mathbf{z}_{b,i} - \hat{\mathbf{z}}^{\mathrm{s}}({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}{}^{-1}\,{}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{W}}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2$$

  Se puede optimizar con métodos no lineales pero requieren de una buena semilla inicial. Es no convexo, solución no única (ambigüedadad de escala)
- Se puede utilizar un enfoque algebraico basado en geometría epipolar para obtener la transformación relativa (a un factor de escala) sin explicitamente computar la posición de los puntos 3D: algoritmos 8-point y 5-point.
- Aplicaciones:
  - Filtrar matches con RANSAC
  - Inicializar una sistemas de SLAM monocular / SfM

# Estimación de movimiento 3D-2D

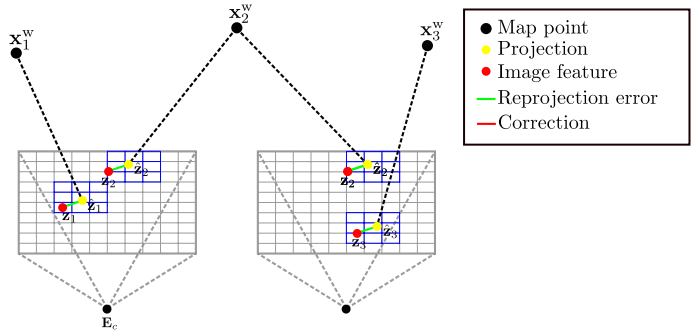
- Dado un conjunto de correspondencias 3D-2D $\{({}^{\mathrm{W}}\mathbf{p}, \mathbf{z}_a)_i\}$ queremos encontrar la pose ${}^{\mathrm{w}}\boldsymbol{\xi}_{\mathrm{a}}$ de la cámara en el mundo.
- Error de reproyección:

$$f({}^{\mathrm{W}}\boldsymbol{\xi}_{\mathrm{a}}) = \sum_{i=1}^{N} \|\mathbf{z}_{a,i} - \hat{\mathbf{z}}({}^{\mathrm{W}}\boldsymbol{\xi}_{\mathrm{a}}{}^{-1}, {}^{\mathrm{W}}\mathbf{p}_i)\|^2$$

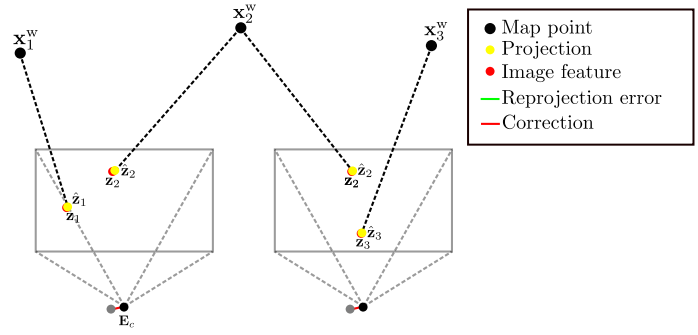
  Se puede optimizar con métodos no lineales pero requieren de una buena semilla inicial. Es no convexo, solución no única (ambigüedad de escala)

- Este problema es conocido como *Perspective-n-Points* (PnP) y existen diferentes enfoques para resolverlo:
  - Direct Linear Transform (DLT)
  - EPnP
  - OPnP
- Aplicaciones:
  - Localización de una cámara dado un mapa de puntos (tracking)

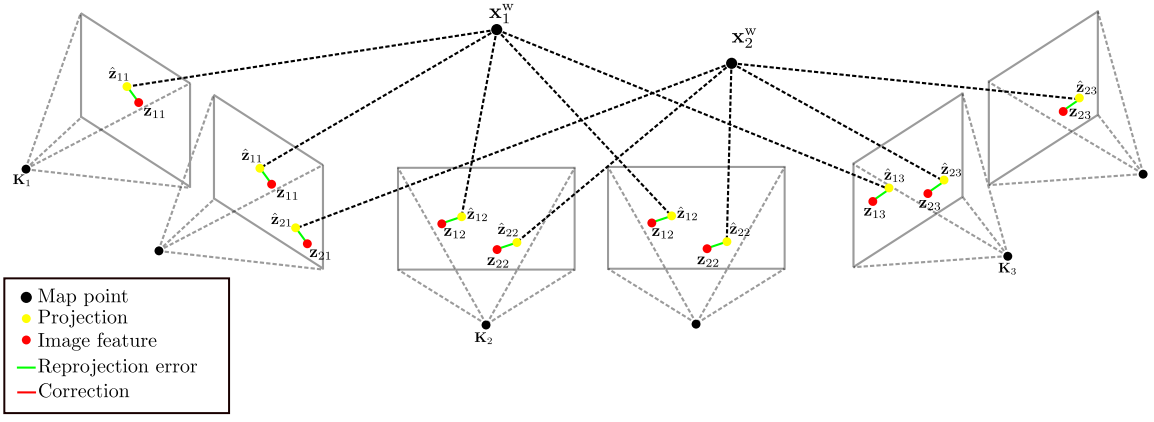
# Matching 3D-2D



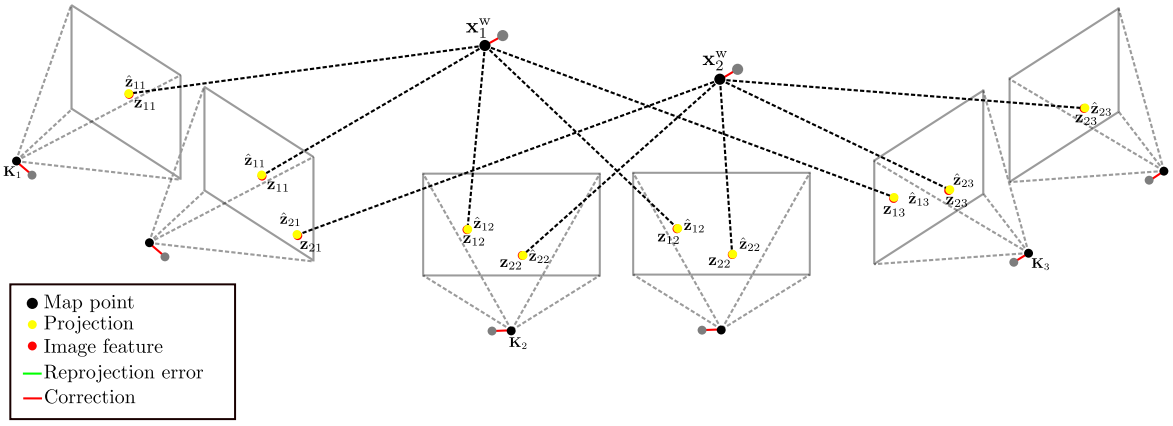
# Ajuste de pose



# Local Bundle Adjustment



# Local Bundle Adjustment



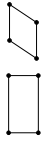
# Estimación de movimiento 3D-3D

- Dado un conjunto de correspondencias 3D en dos sistemas de coordenadas distintos: $\{({}^{\mathrm{a}}\mathbf{p}, {}^{\mathrm{b}}\mathbf{p})_i\}$ queremos encontrar la transformación relativa ${}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}$.
- Error geométrico 3D:

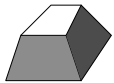
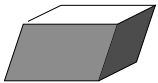
$$f({}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}}) = \sum_{i=1}^{N} \| {}^{\mathrm{a}}\mathbf{p}_i - {}^{\mathrm{a}}\boldsymbol{\xi}_{\mathrm{b}} \, {}^{\mathrm{b}}\mathbf{p}_i \|^2$$

  Corresponde a alineamiento de nube de puntos por mínimos cuadrados.
- Solución cerrada, eg.g Arun et al, 1987
- Aplicaciones:
  - Obtención de movimiento relativo para cámaras estéreo (mediante el uso de puntos triangulados) o RGB-D (mediciones con profundidad)
  - Corrección de Loop Closure (variante con estimación de escala para SLAM monocular)

# Tipos de transformaciones 2D

| Group | Matrix | Distortion | Invariant properties |
|---|---|---|---|
| Projective<br>8 dof | $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ |  | Concurrency, collinearity, **order of contact**: intersection (1 pt contact); tangency (2 pt contact); inf ections (3 pt contact with line); tangent discontinuities and cusps. cross ratio (ratio of ratio of lengths). |
| Aff ne<br>6 dof | $\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (e.g. midpoints), linear combinations of vectors (e.g. centroids). The line at inf nity, $l_\infty$ . |
| Similarity<br>4 dof | $\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Ratio of lengths, angle. The circular points, $I, J$ (see section 2.7.3). |
| Euclidean<br>3 dof | $\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$ |  | Length, area |

# Tipos de transformaciones 3D

| Group | Matrix | Distortion | Invariant properties |
|---|---|---|---|
| Projective<br>15 dof | $\begin{bmatrix} \mathsf{A} & \mathsf{t} \\ \mathsf{v}^\mathsf{T} & v \end{bmatrix}$ |  | Intersection and tangency of surfaces in contact. Sign of Gaussian curvature. |
| Affine<br>12 dof | $\begin{bmatrix} \mathsf{A} & \mathsf{t} \\ \mathsf{0}^\mathsf{T} & 1 \end{bmatrix}$ |  | Parallelism of planes, volume ratios, centroids. The plane at infinity, $\pi_\infty$, (see section 3.5). |
| Similarity<br>7 dof | $\begin{bmatrix} s\mathsf{R} & \mathsf{t} \\ \mathsf{0}^\mathsf{T} & 1 \end{bmatrix}$ |  | The absolute conic, $\Omega_\infty$, (see section 3.6). |
| Euclidean<br>6 dof | $\begin{bmatrix} \mathsf{R} & \mathsf{t} \\ \mathsf{0}^\mathsf{T} & 1 \end{bmatrix}$ |  | Volume. |

# Calibración de cámara

**Clase Calibración por Cyrill Stachniss:** https://www.youtube.com/watch?v=-9He7Nu3u8s

# Material utilizado para estas slides

- Slides fuertemente basadas de `https://youtu.be/ebMyBbkkHWk`
- `https://3d.bk.tudelft.nl/courses/geo1016/slides/Lecture_03_Calibration.pdf`

# Bibliografía

[1] Richard Hartley y Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2.ª ed. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521540518. DOI: 10.1017/CBO9780511811685.