Visión artificial y Robótica
Geometría

Depto. de Ciencia de la Computación e Inteligencia Artificial

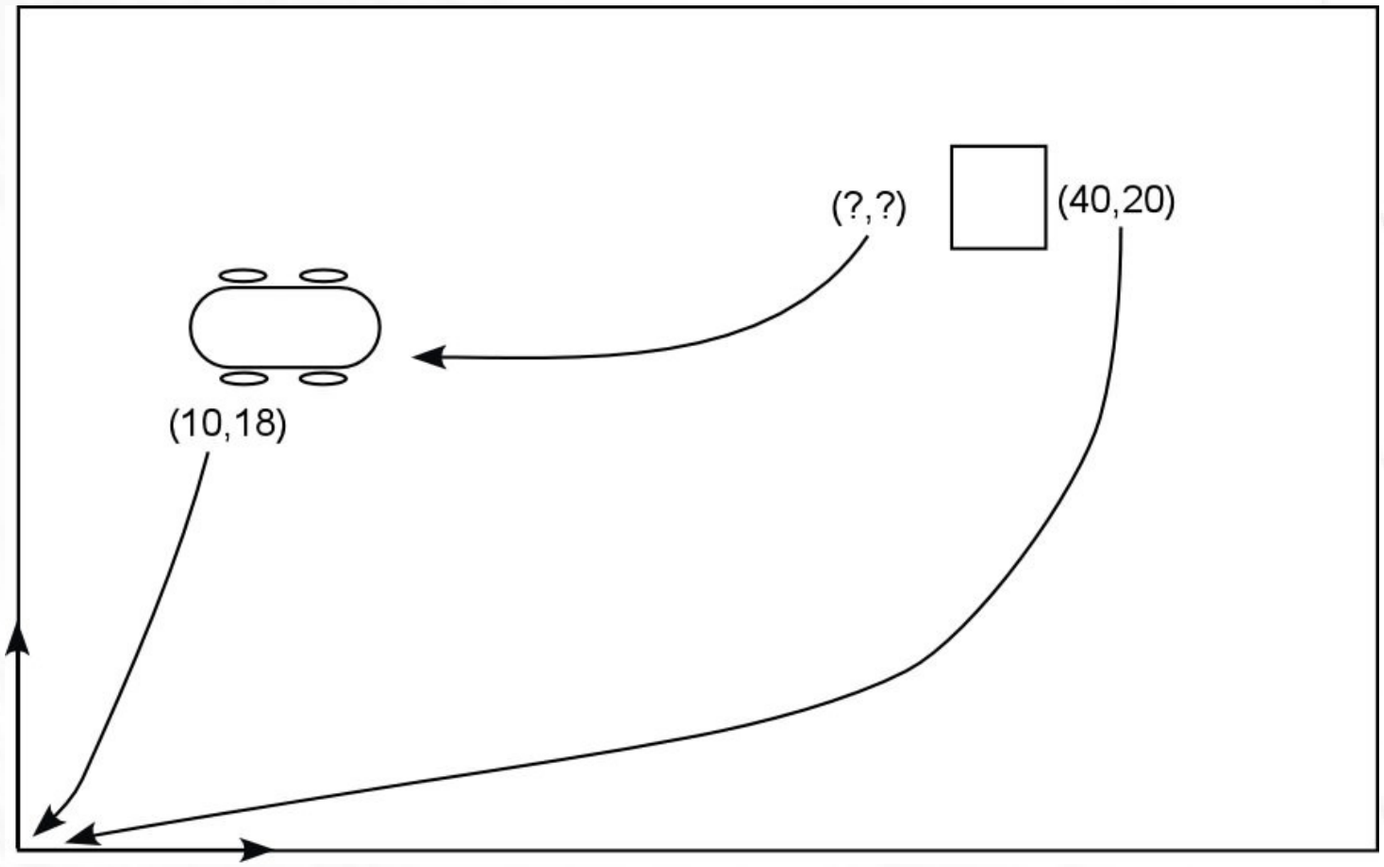
Contenidos

- Geometría 2D y 3D
- Transformación de coordenadas
- Calibración de la cámara

Álgebra necesaria

Necesitamos herramientas geométricas para manejar las posiciones de los robots y de los objetos en el espacio

- Posición del robot dentro de un entorno
- Posición de un objeto dentro del mismo entorno
- Posición relativa del objeto con respecto al robot



Dimensiones utilizadas

Podemos tener un espacio de dos o tres dimensiones

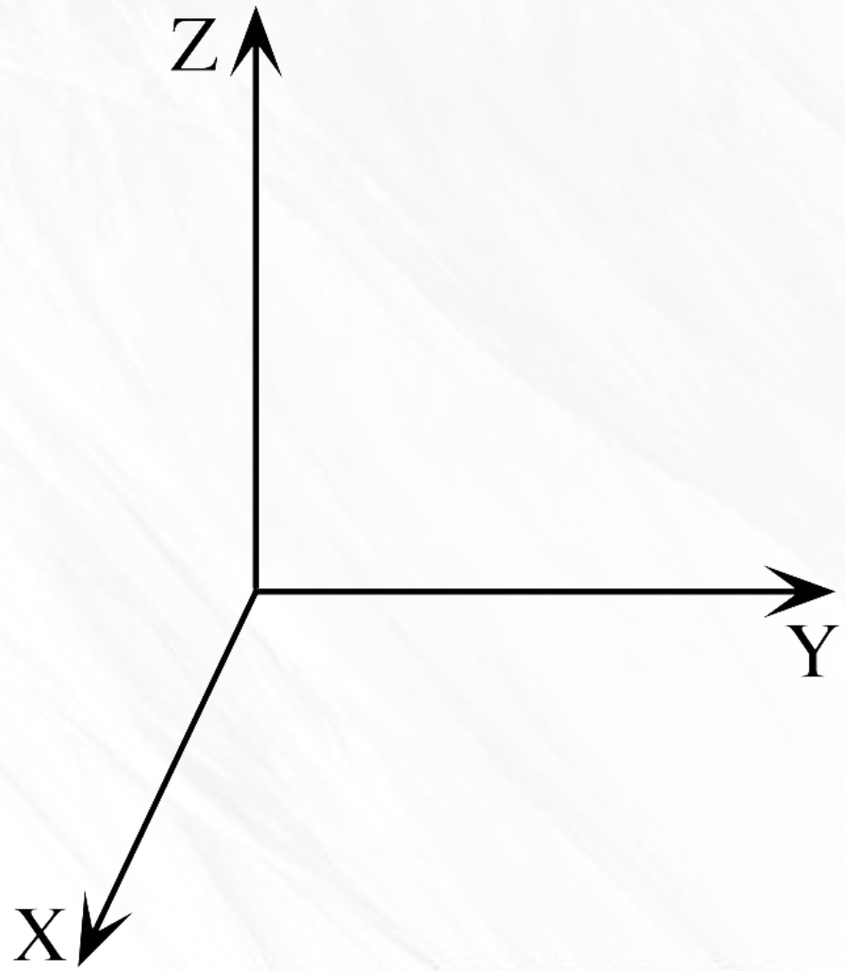
- Veremos el caso general en tres dimensiones
- Restringiremos a dos dimensiones más el ángulo

El origen de un sistema de coordenadas se puede colocar en cualquier posición

Utilizaremos más de un sistema de coordenadas

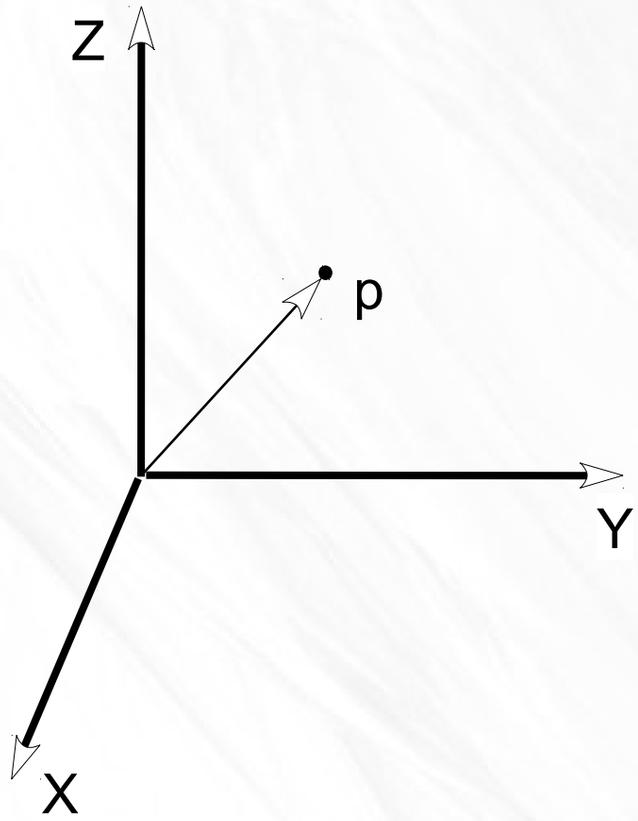
Etiquetado de los ejes de coordenadas

Regla de la mano derecha



Coordenadas cartesianas

Un punto en el espacio se define mediante las coordenadas de su posición con respecto al origen del sistema



$$\vec{p} = [x\ y\ z]^T$$

$$\vec{p}(x_1, y_1, z_1)$$

$$\vec{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Coordenadas homogéneas

Surgen en gráficos para solucionar la capacidad de representación de los números en un ordenador

También permiten poder disponer de una sola matriz para transformación de coordenadas

$$[x\ y\ z]^T \longrightarrow [w\,x\ w\,y\ w\,z\ w\,]^T$$

Grados de libertad

Grado de libertad: cada uno de los movimientos (desplazamiento y/o rotación) que se pueden realizar

Un cuerpo 3D tiene 6 grados de libertad (dof: degree of freedom): 3 rotaciones y 3 posiciones

Un cuerpo 2D tiene 3 dof: 1 rotación y 2 posiciones

Matrices de transformación

Nos permiten transformar las coordenadas de un punto

$$T = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

r_i=términos de rotación

t_i=términos de traslación

Definen un conjunto de transformaciones (rotaciones y traslaciones)

Matrices de transformación

Matriz de translación:

$$Tras(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrices de rotación:

$$Rot^x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

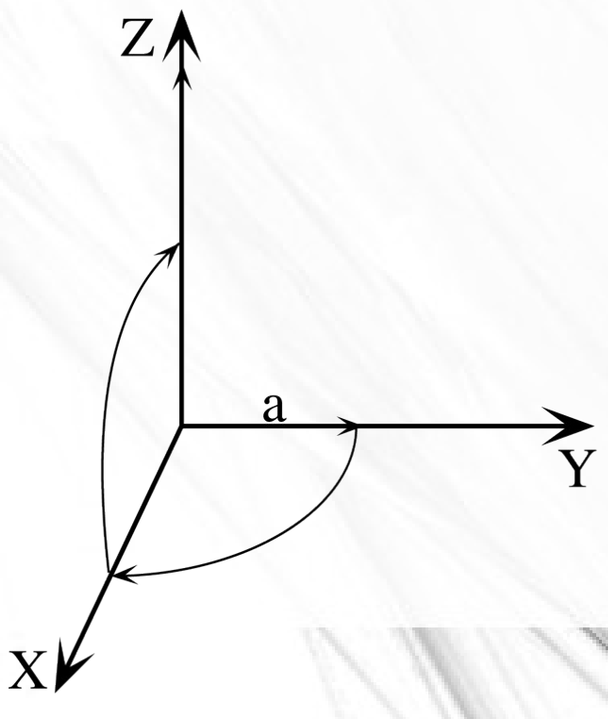
$$Rot^y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

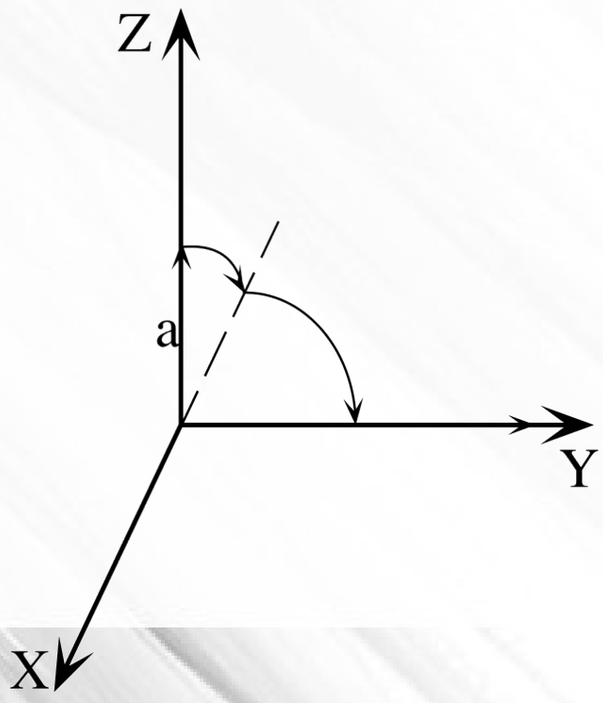
$$Rot^z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrices de transformación II

Cuando queremos realizar varias trasformaciones sucesivas multiplicamos las matrices de transformación correspondientes

Las matrices se aplican siempre de derecha a izquierda. El resultado es distinto si no aplicamos el sentido correcto





Ejemplo

- Trasladamos el punto una distancia a lo largo del eje Y.
- Rotamos con respecto al eje Z -90 grados.
- Rotamos con respecto al eje Y -90 grados.
- Trasladamos una distancia a lo largo del eje Z.

Matriz resultante

$$
\begin{aligned}
T &= Tras(0,0,a)\,Rot^{y}(-90)\,Rot^{z}(-90)\,Tras(0,a,0) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(-90) & 0 & \sin(-90) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-90) & 0 & \cos(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\cdot \begin{bmatrix} \cos(-90) & -\sin(-90) & 0 & 0 \\ \sin(-90) & \cos(-90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Resultado

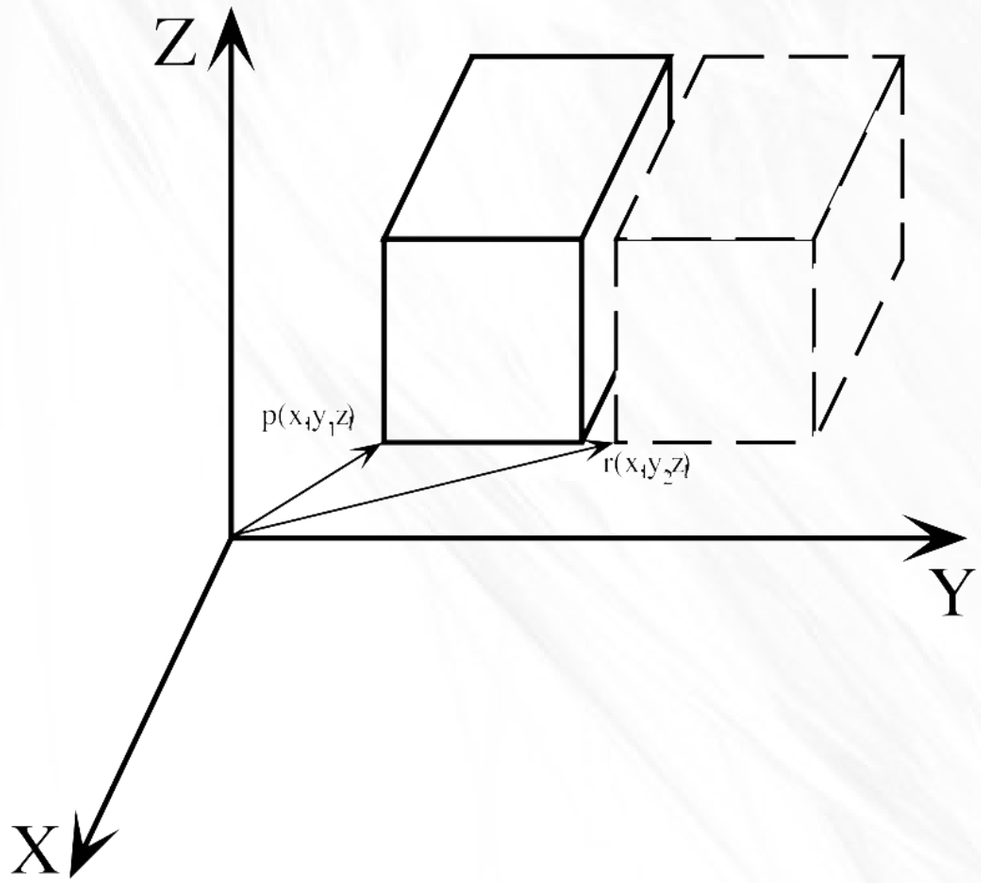
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}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

$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & a \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ejemplo: transformad el punto $[0\ 0\ 0\ 1]^T$

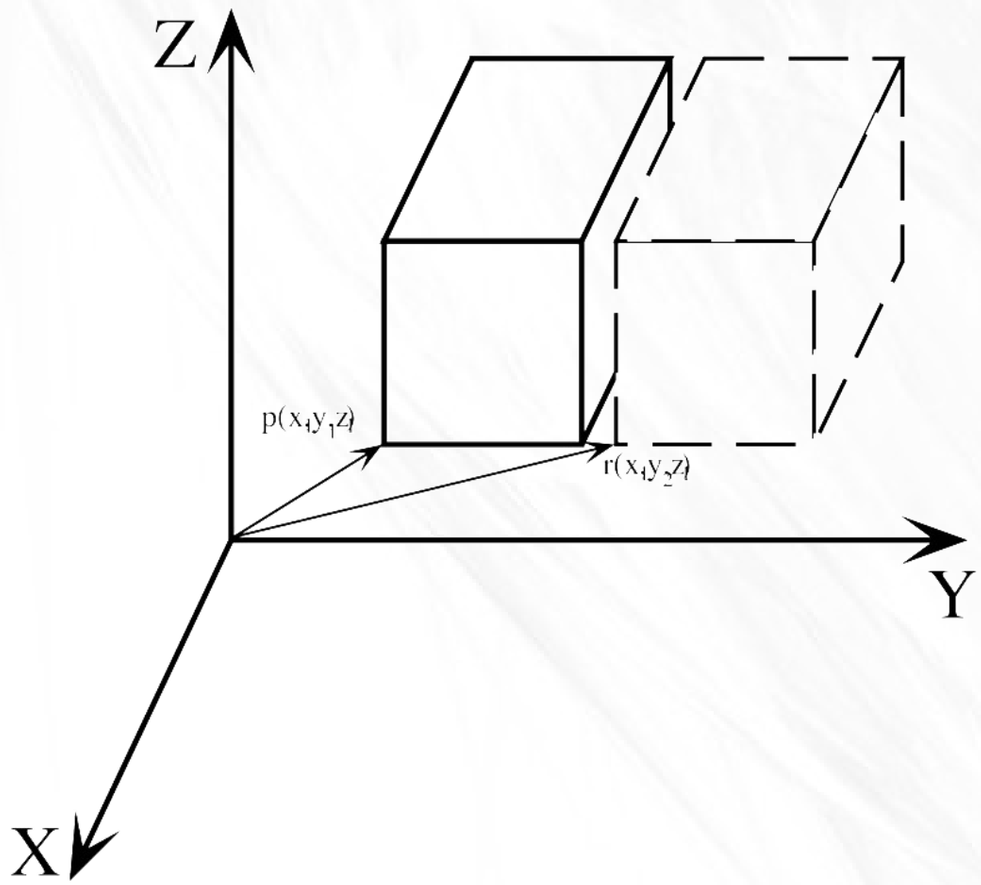
Localización de objetos

Objeto definido por cada uno de sus vértices



Localización de objetos

Objeto definido por cada uno de sus vértices



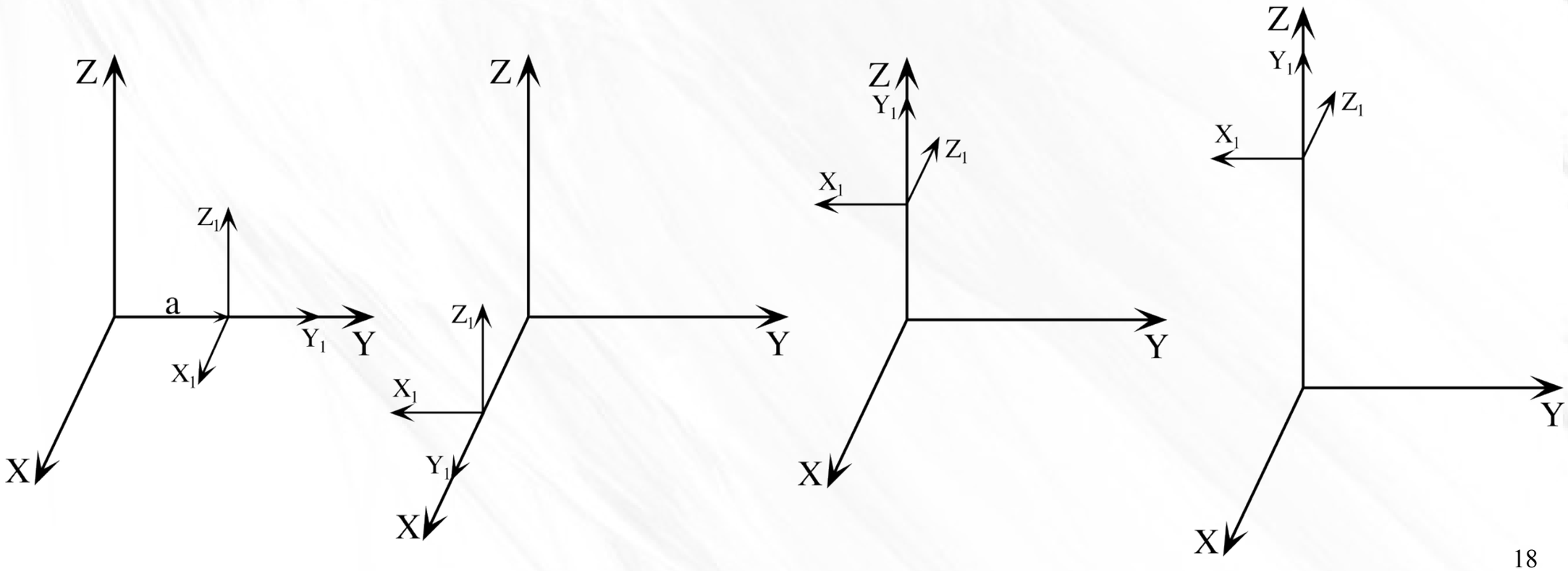
Alternativa: Definir un nuevo sistema de coordenadas en el objeto. Las coordenadas de los vértices del objeto se definen con respecto al nuevo sistema.

Relación entre los sistemas de coordenadas

- Vamos a tener tantos sistemas de coordenadas como necesitemos
- Un sistema de coordenadas tiene una posición x, y, z y una rotación para cada eje
- Tendremos que ser capaces de relacionar todos los sistemas de coordenadas
- Dispondremos de una matriz de transformación que relaciona dos sistemas

Ejemplo II

Aplicación de la misma transformación anterior a un sistema de coordenadas:



Transformación de coordenadas

Disponemos de las coordenadas de un punto con respecto a un sistema. Para encontrar las coordenadas de ese punto con respecto a otro sistema de coordenadas, multiplicamos las coordenadas del punto por la matriz que relaciona ambos sistemas

Transformación de coordenadas

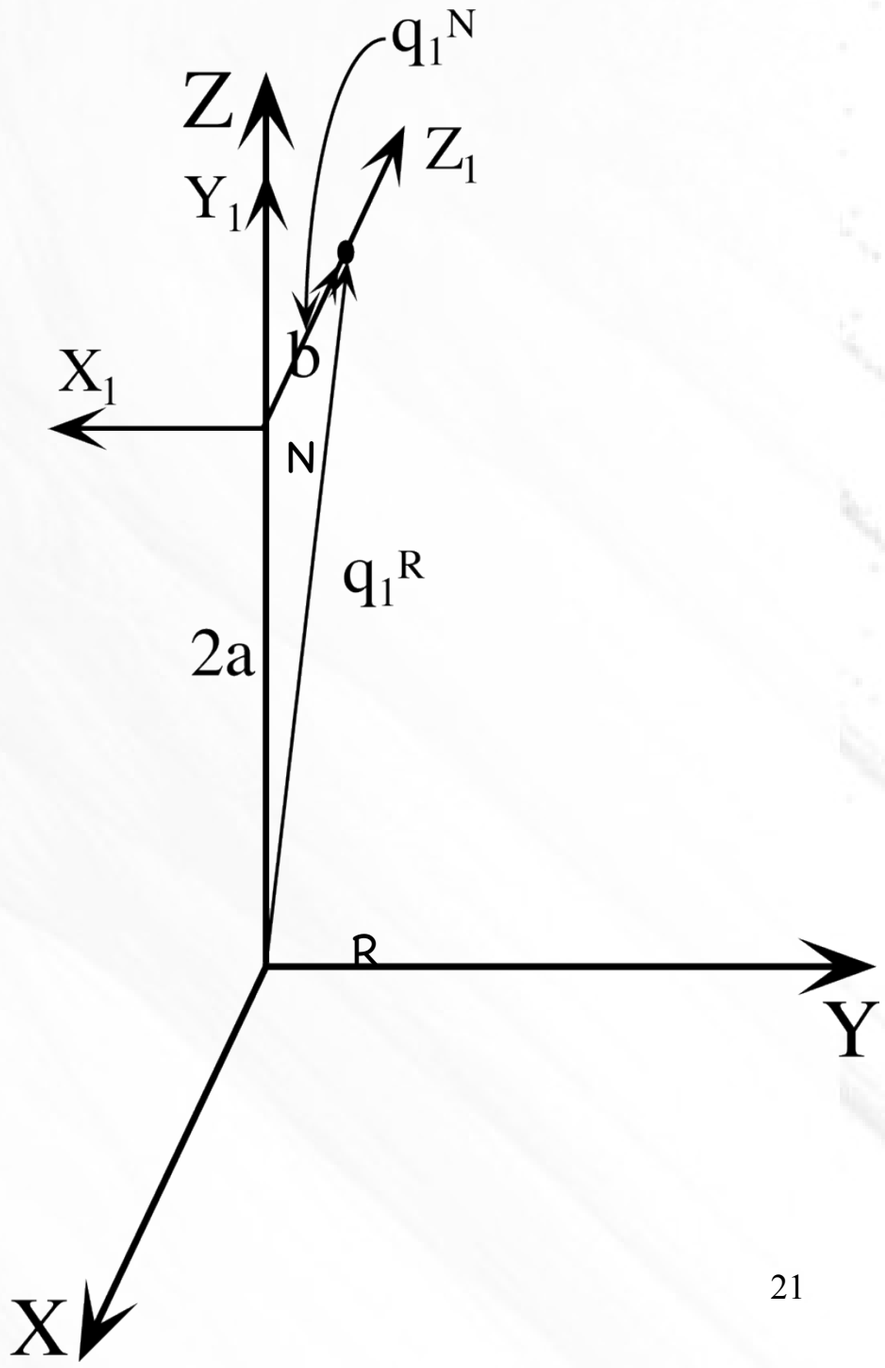
- Tenemos el punto $q_1^N = [0\ 0\ b\ 1]^T$
- La matriz que nos relaciona los sistemas es la anterior

- Las nuevas coordenadas se calculan:

$$q_1^R = T_N^R \cdot q_1^N = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2a \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} -b \\ 0 \\ 2a \\ 1 \end{bmatrix}$$

Transformación

$$q_1^R = T_N^R \cdot q_1^N = \begin{bmatrix} -b \\ 0 \\ 2a \\ 1 \end{bmatrix}$$



Transformación inversa

Ahora queremos calcular las coordenadas del punto antes calculado con respecto al sistema N:

$$(T_N^R)^{-1} q_1^R = (T_N^R)^{-1} T_N^R q_1^N = q_1^N$$

Construcción de la matriz inversa

$$T = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T^{-1} = \begin{bmatrix} x_x & x_y & x_z & -\vec{p} \cdot \vec{x} \\ y_x & y_y & y_z & -\vec{p} \cdot \vec{y} \\ z_x & z_y & z_z & -\vec{p} \cdot \vec{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde $\vec{x} = [x_x\ x_y\ x_z]^T$, $\vec{y} = [y_x\ y_y\ y_z]^T$, $\vec{z} = [z_x\ z_y\ z_z]^T$ y $\vec{p} = [p_x\ p_y\ p_z]^T$.

Cálculo de la matriz inversa

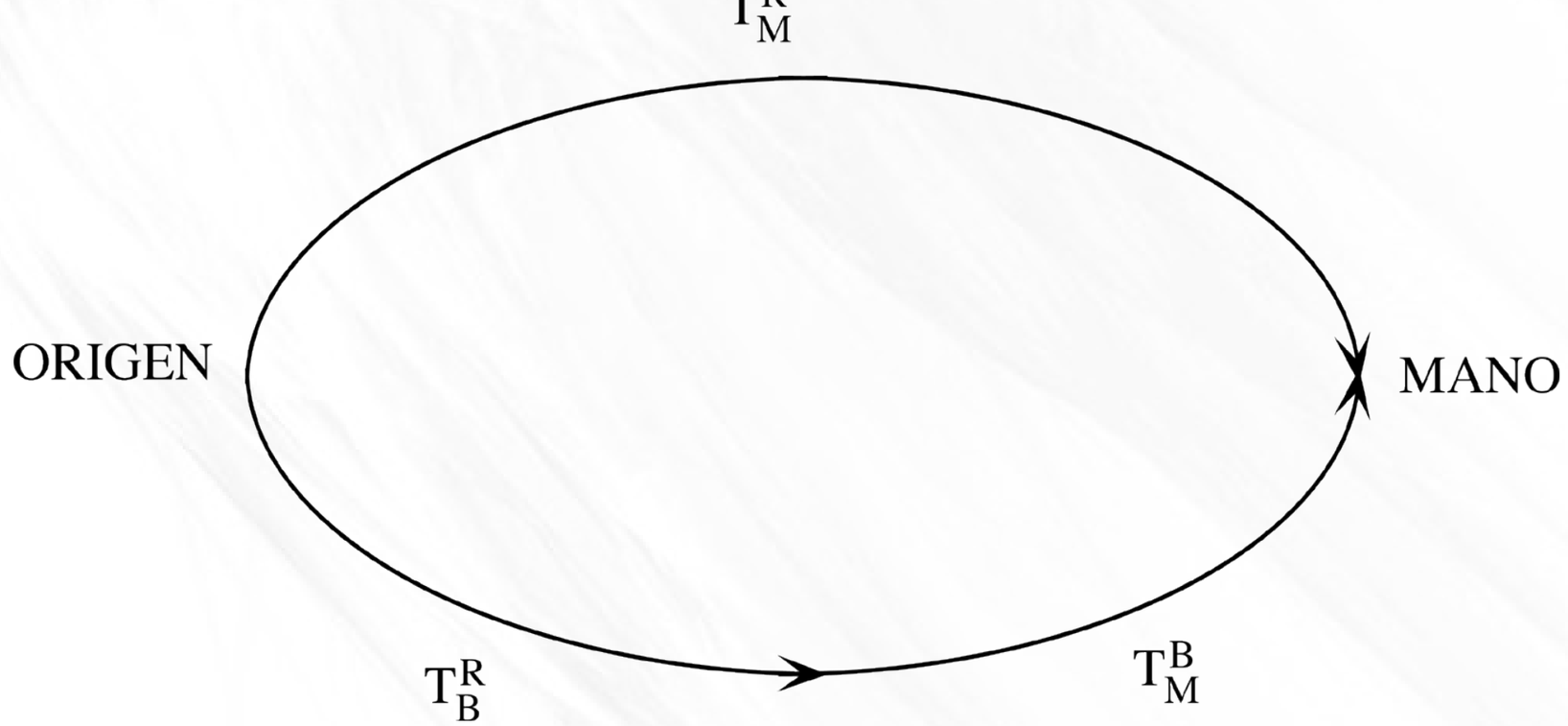
- Cálculo de la matriz $(T_N^R)^{-1}$

$$T_N^R = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2a \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (T_N^R)^{-1} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -2a \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Calcular el punto q_1^N

Grafos de transformación

Permiten las transformaciones entre distintos sistemas de coordenadas



Coordenadas en 2D

Tenemos un mundo en dos dimensiones (x, y) y una orientación (θ) (tres grados de libertad)

Las tres coordenadas (x, y, θ) definen tanto una posición como un sistema de coordenadas

Matriz de rotación:

$$Rot(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

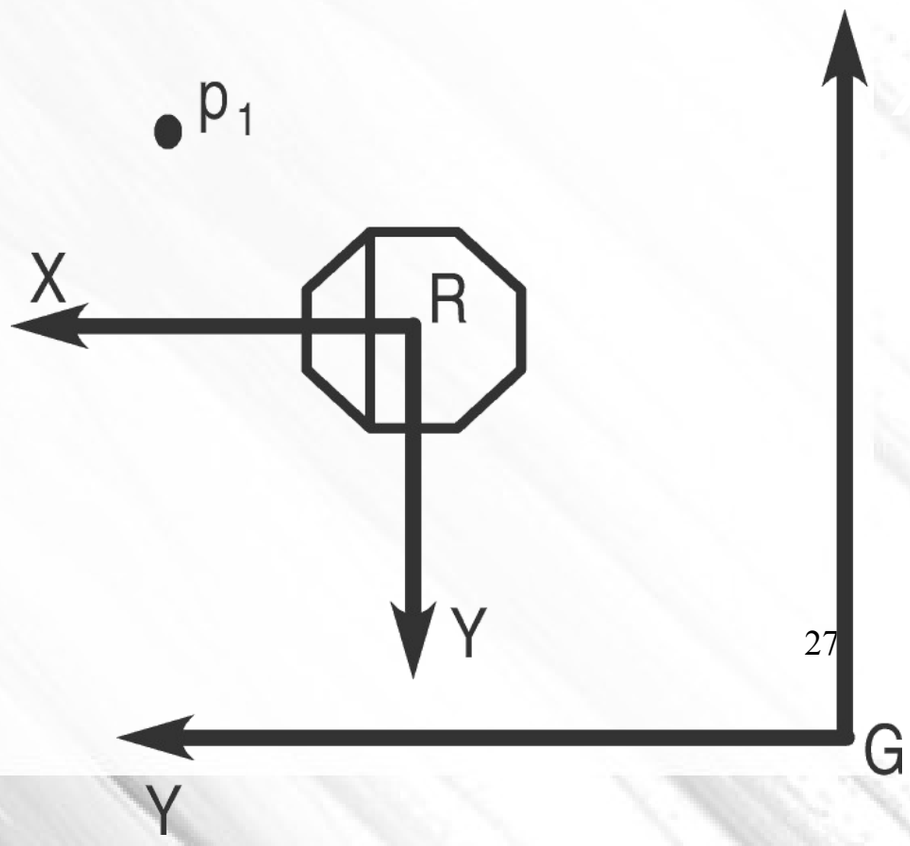
$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Sistemas de coordenadas 2D

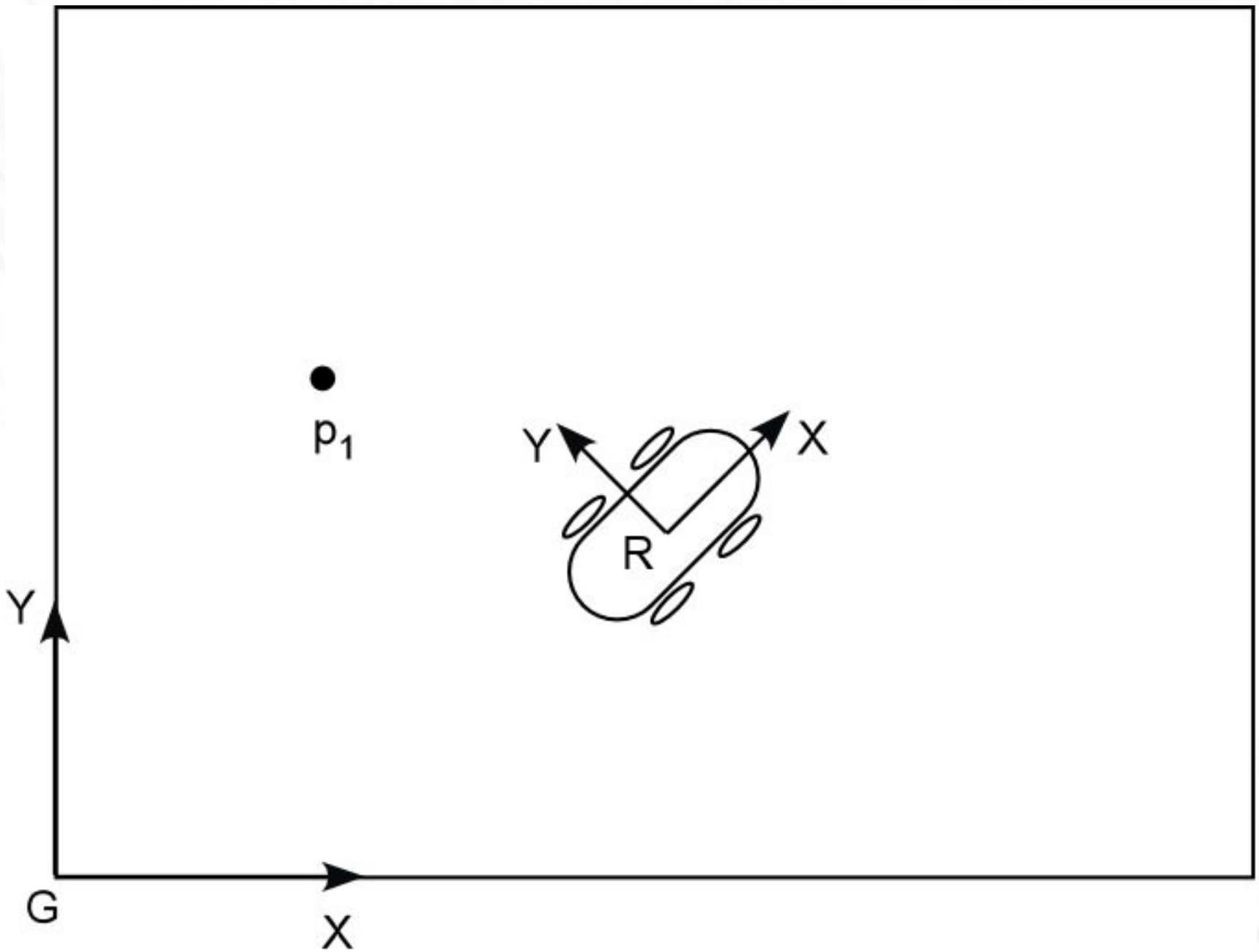
Vamos a tener dos sistemas principales: el global y el local

Las coordenadas del robot cambian cuando se desplaza

Debemos ser capaces de calcular las coordenadas de los objetos en nuestro entorno



Relación de los sistemas de coordenadas



Transformación de coordenadas

Sistema de coordenadas del robot:

$$R = [x_r \; y_r \; \theta_r]^T$$

Coordenadas del punto p_1 con respecto a R

$$p_1^R = [x_1 \; y_1 \; \theta_1]^T$$

Coordenadas del punto con respecto a G

$$P_1^G = Rot(-\theta_r) p_1^R + \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix}$$

Ejemplo

Tenemos el punto

$$p_1^R = [5\ -3\ 0]^T$$

El sistema del robot

$$R = [3\ 4\ 90]^T$$

$$P_1^G = Rot(-90)p_1^R + R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ -3 \\ 0 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \\ 90 \end{bmatrix} = \begin{bmatrix} 6 \\ 9 \\ 90 \end{bmatrix}$$

Transformación de coordenadas inversa

Coordenadas del punto con respecto a R conocidas las de G:

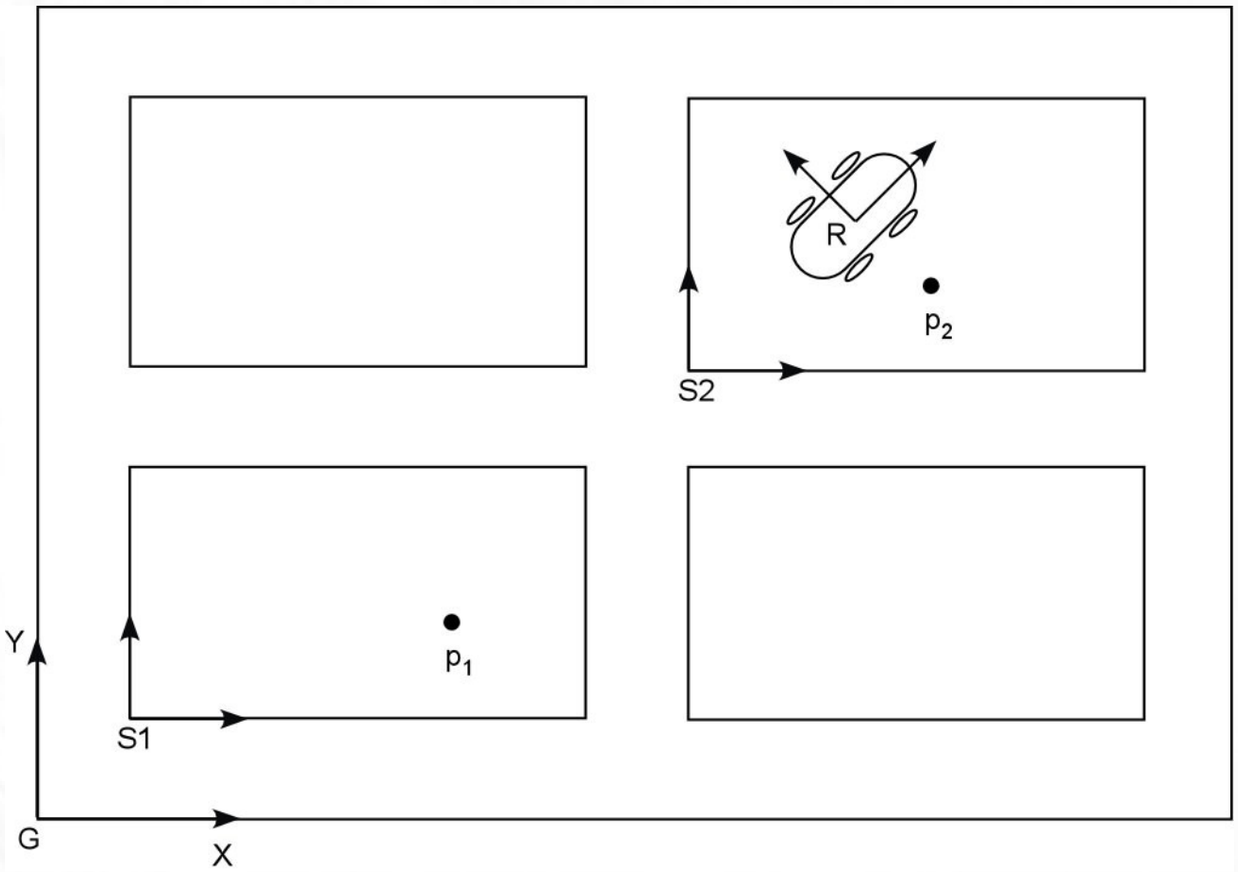
$$P_1^G = Rot(-\theta_r)p_1^R + \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix}$$

$$Rot(-\theta_r)p_1^R = P_1^G + \begin{bmatrix} -x_r \\ -y_r \\ -\theta_r \end{bmatrix}$$

$$p_1^R = Rot(-\theta_r)^{-1}\left(P_1^G + \begin{bmatrix} -x_r \\ -y_r \\ -\theta_r \end{bmatrix}\right)$$

$$p_1^R = Rot(\theta_r)\left(P_1^G + \begin{bmatrix} -x_r \\ -y_r \\ -\theta_r \end{bmatrix}\right)$$

Ejercicio



$S_1^G=[100\ 100\ 0]^T$ $S_2^G=[600\ 400\ 0]^T$

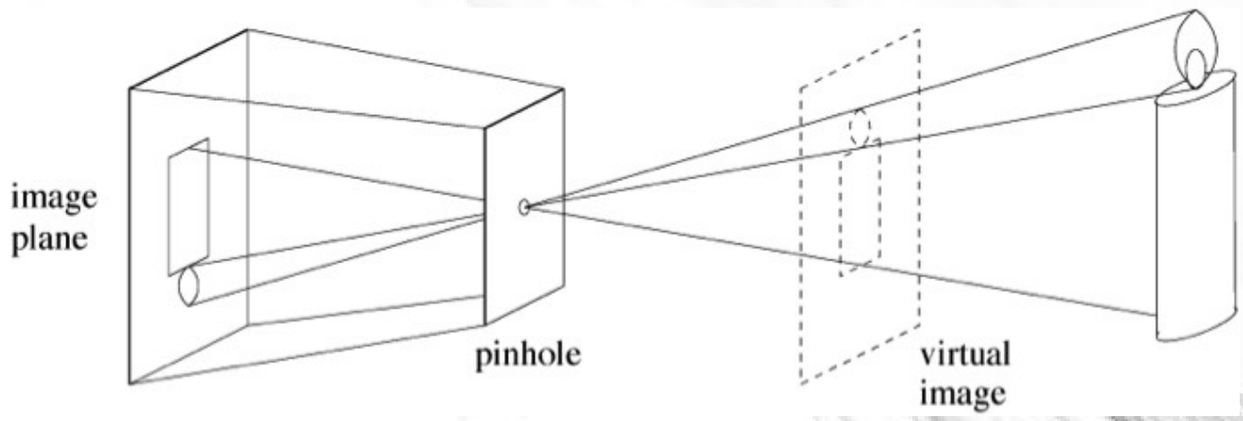
$R^{S2}=[100\ 100\ 45]^T$ $P_1^{S1}=[250\ 50\ 0]^T$

$P_2^G=[735\ 465\ 0]^T$

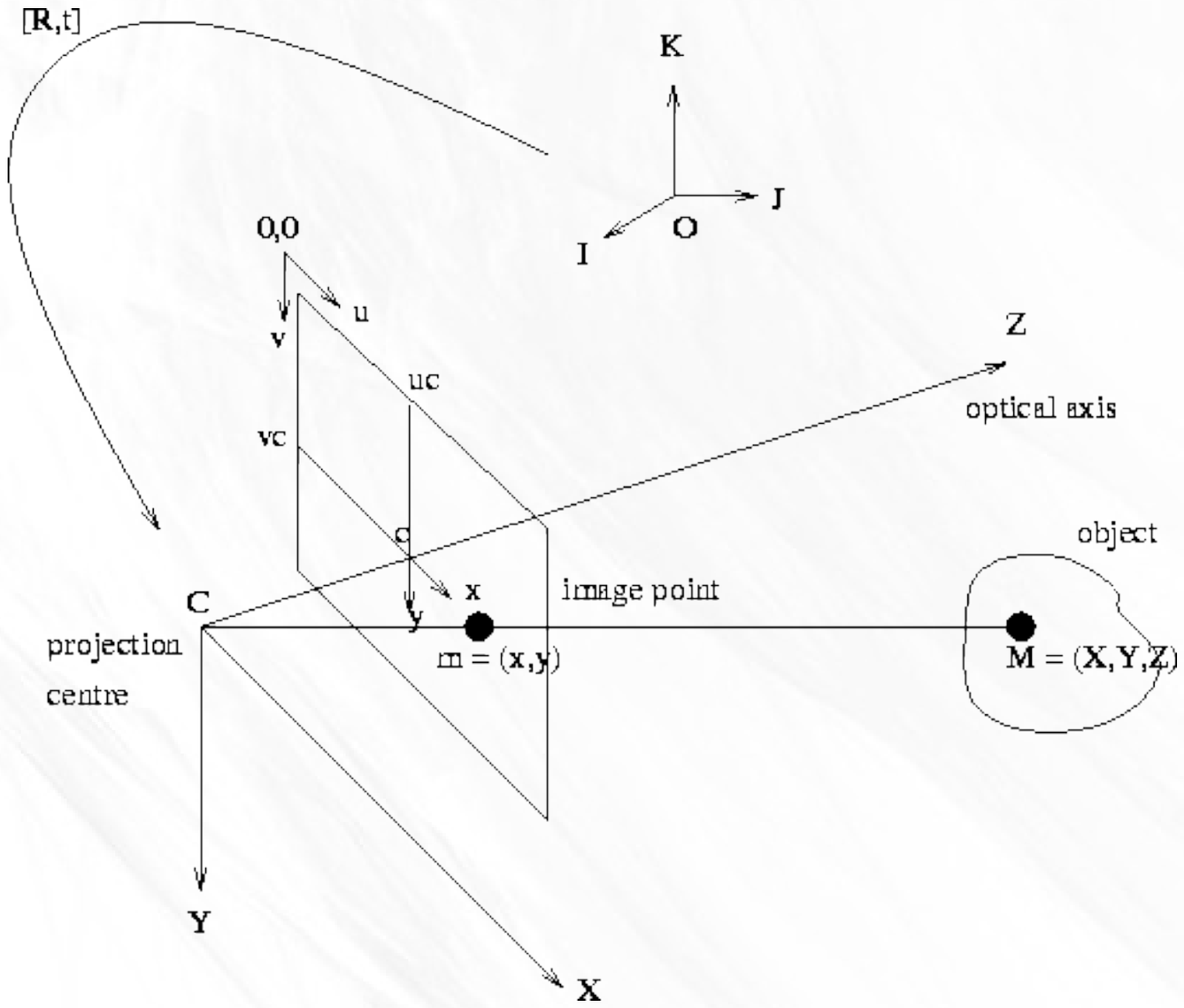
Calcular: R^G; P_1^R; P_2^{S2}; P_2^R;

Calibración de la cámara

- Proceso de captura de una imagen con una cámara *pinhole*:
  - Un punto en el espacio se proyecta a través de la focal de la cámara en el plano imagen (ejemplo)
  - Partimos de un punto con 3 coordenadas X,Y,Z y nos quedamos con solo 2 x,y
- Para ser capaces de recuperar la información 3D, debemos ser capaces de saber cómo se produce la proyección
- Veremos el caso sencillo, pero hay que tener en cuenta otros factores como lentes, distorsión,etc.

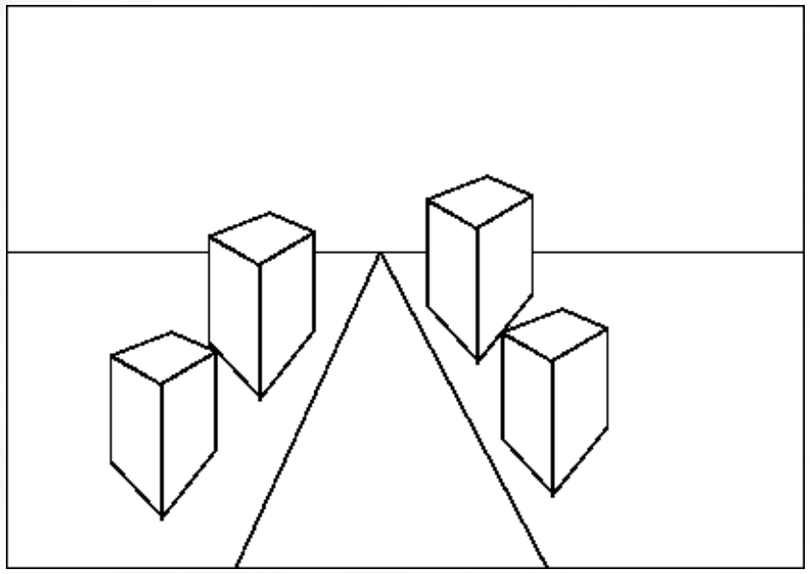


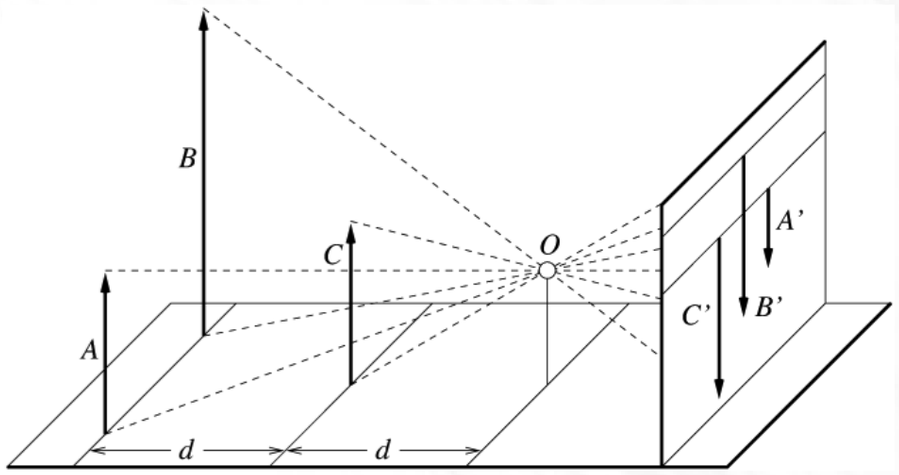
Coordenadas de la calibración



Efectos de la proyección

Proyección en perspectiva, aunque existen otras proyecciones como la ortográfica





Proyección de los puntos

- Se producen dos proyecciones
  - Por parámetros externos: posición de la cámara en el espacio.
  - Por parámetros internos: los propios de la cámara (focal, distorsiones, etc.)

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I.i} & \mathbf{J.i} & \mathbf{K.i} & T_x \\ \mathbf{I.j} & \mathbf{J.j} & \mathbf{K.j} & T_y \\ \mathbf{I.k} & \mathbf{J.k} & \mathbf{K.k} & T_z \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix}$$

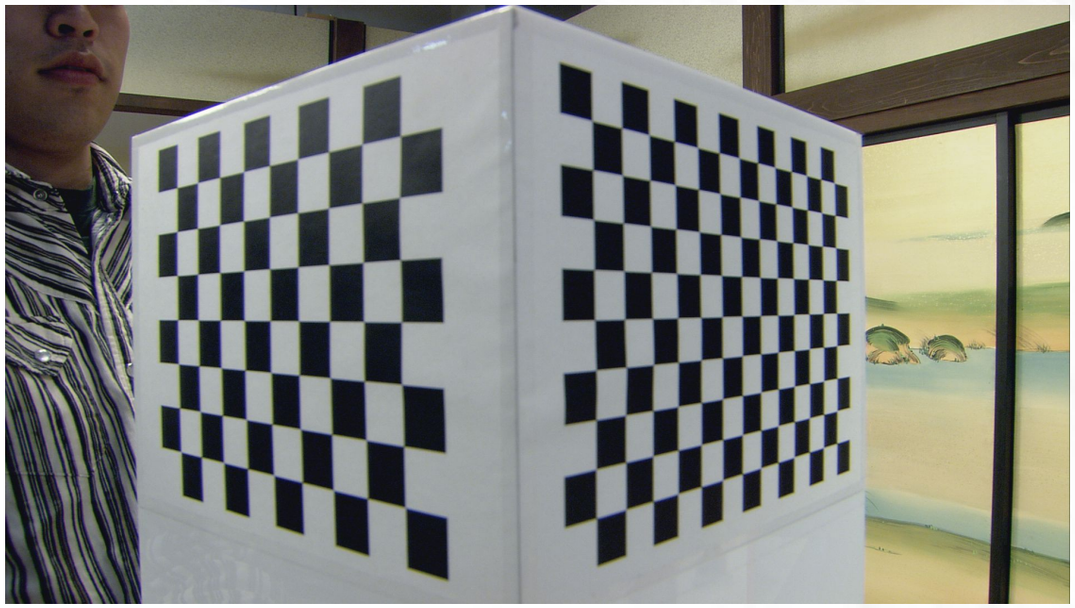
¿Qué nos da la calibración?

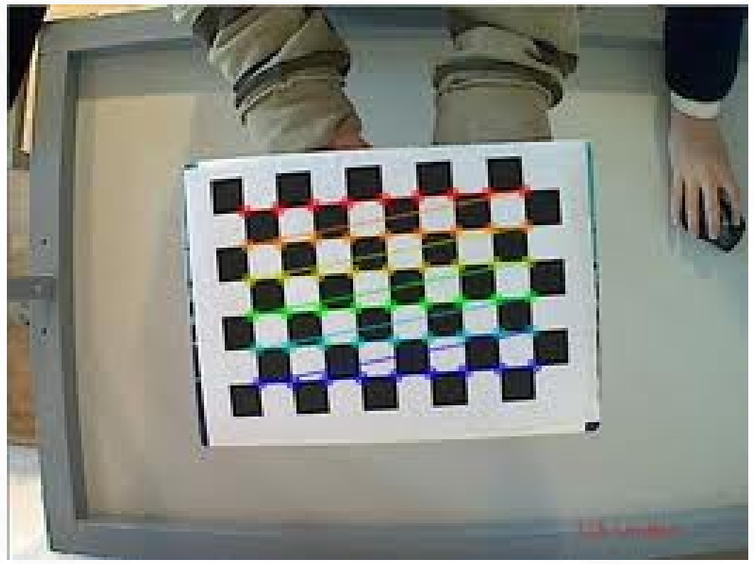
- Si calibramos una cámara, lo que se nos proporciona es una posible línea donde puede estar el punto (pensad en ello un momento)

- ¿Entonces cómo calibramos?

Correspondencias

- Hay que encontrar los puntos en la imagen que se corresponden con los puntos 3D
- Se puede usar un *tablero de ajedrez* para encontrar esas correspondencias





Correspondencias

- El uso del tablero permite obtener de manera automática los puntos esquina de los cuadrados
- Debemos pasar a la calibración las dimensiones reales del cuadrado
- Con esto, vamos a tener 11 variables desconocidas (ver transparencia 36), pero podemos hacer un sistema de ecuaciones

Sistema de ecuaciones

- Normalmente capturamos varias imágenes y determinamos las correspondencias para todas ellas
- El objetivo es colocar el tablero cada vez en una posición, cambiando su orientación y posición.
- ¿Cómo se puede resolver el sistema?

Resolución del sistema

- Método de Tsai
  http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/
- Transformación linear directa (usa SVD)
- Minimización no linear
- Lo mejor, usar OpenCV o Matlab :-)

Otras técnicas

- Imaginad que una cámara en marte se descalibra :-s (descalibrar puede ser que haya perdido el foco, que la lente se haya movido, etc.)
- ¿Cómo calibrarías?

Referencias

- Computer Vision: A Modern Approach. David A. Forsyth, Jean Ponce.
- Transparencias de ese libro