

Práctica 2. POSE: Posición y Orientación a partir de Marcas Codificadas

1. Objetivo

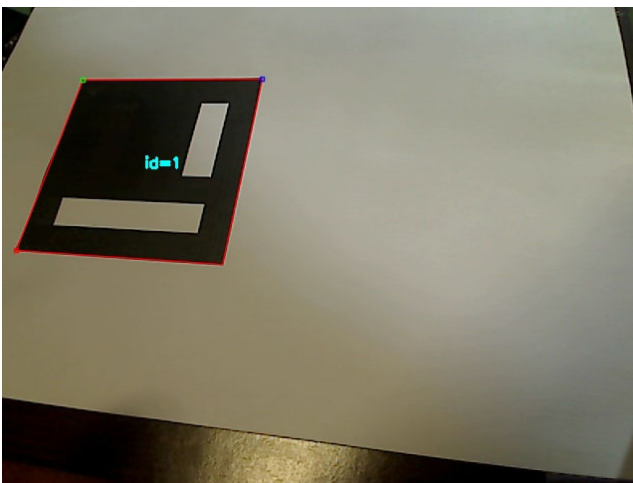
El objetivo de la práctica es que el alumno sea capaz de calcular la posición y orientación de una marca plana codificada, a partir de la homografía del plano y los datos de calibración de la cámara.

2. Escenarios

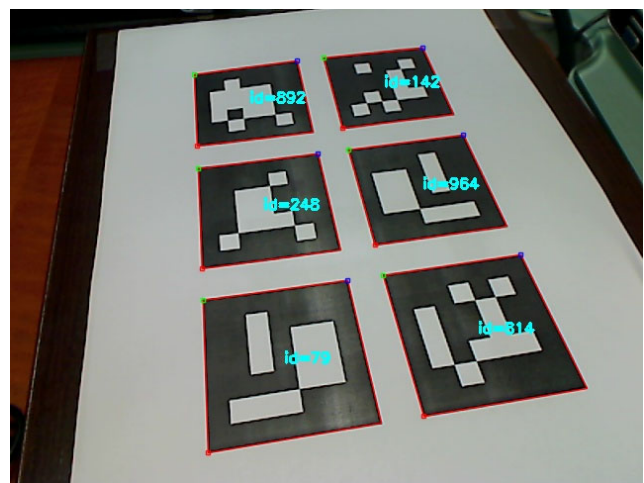
Se trabajará con imágenes capturadas por una cámara USB conectada al ordenador. Para ello se utilizará el software proporcionado en la documentación de la práctica para la captura y detección de marcas codificadas.

Se proporcionan dos ficheros pdf para imprimir una hoja con las marcas codificadas:

- Escenario1: imagen con una sola marca id=1
- Escenario 2: imagen con 6 marcas dispuestas en una matriz de 2x3 con diferentes identificadores



Escenario 1



Escenario 2

3. Software utilizado

El desarrollo de la práctica se realizará en Matlab. Para ello se utiliza un programa para la detección de marcas desarrollado en OpenCV (<http://opencv.org>) usando la librería ARUCO. El código fuente se encuentra disponible en el blog de la asignatura (<http://umh1782.edu.umh.es/opencv/>).

Programas suministrados:

- **markers.m:** realiza una llamada al ejecutable que detecta las marcas en la imagen pasada como parámetro o bien procedente de una cámara USB. Devuelve los datos con el identificador de marcas y la posición 2D de las esquinas detectadas y la imagen capturada. Uso:
 - Sin parámetros: procesa en tiempo real de una cámara USB
`[id, points, im] = markers;`
 - Con Parámetros de entrada: procesa un fichero de imagen
`[id, points, im] = markers( file, ~ );`
 - file: Nombre del fichero con la imagen a procesar
 - Parámetro extra: si añadimos un segundo parámetro (cualquier valor) activa la visualización de la detección de las marcas durante unos segundos

Resultados devueltos

- id: vector con los identificadores de marca detectados.
 - points: array de tres dimensiones (j,k,i):
 - j: índice de las dos coordenadas (c,f)
 - k: índice de las 4 esquinas del marcador
 - i: es el índice en el vector de ids
 - im: matriz con la imagen capturada u cargada desde el fichero.
- **markers.exe:** ejecutable para calcular las marcas (en las versiones de linux o Mac es un ejecutable sin extensión). No es necesario ejecutarlo externamente, ya que se es preferible usar el script Matlab markers.m
 - **showmarkers.m:** muestra las esquinas de la marca y el identificador sobre la imagen. Parámetros:
`showmarkers(id, points, im)`
 - **comp_distortion_oulu.m:** función para corregir la distorsión (Matlab Calibration Toolbox)
 - **pose1.m:** esquema programa solución para la práctica con el escenario 1
 - **pose2.m:** esquema programa solución para la práctica con el escenario 2
 - **homografia.m:** esquema programa para cálculo de la homografía entre planos proyectivos

Otros ficheros:

- **marker_id1.pdf:** fichero con la imagen de una marca para imprimir
- **markersBoard.pdf:** fichero con la imagen de un conjunto de marcas para imprimir
- **calib_LogitechC300.m:** fichero con los datos de calibración de la cámara Logitech C300
- **calib_LogitechWP9000.m:** fichero con los datos de calibración de la cámara Logitech WP9000
- **calib_Trust.m:** fichero con los datos de calibración de la cámara Trust
- **board_data.m:** fichero con los datos del markersBoard.pdf

- **marker_data.m:** fichero con los datos del marker_id1.pdf

4. Tareas

A continuación se detallan las etapas de las que se compone la ejecución de la práctica. En cada una de ellas se describe una consideración sobre su carácter (obligatoria, opcional o recomendable).

Los ficheros que deberá editar y completar el alumno son:

- **pose1.m**: esquema programa solución para la práctica con el escenario 1
- **pose2.m**: esquema programa solución para la práctica con el escenario 2
- **homografia.m**: esquema programa para cálculo de la homografía entre planos proyectivos

Asimismo el alumno deberá crear una función para dibujar los ejes de coordenadas 3D en el ejercicio opcional:

- **axis3D.m**: programa solución para dibujar los ejes de coordenadas 3D

4.1. Calibración de la cámara (Opcional)

Se debe disponer del fichero de calibración de la cámara obtenido en la práctica 1. Si no se dispone del mismo, entre los ficheros suministrados se incluyen los datos de calibración de varios modelos de cámaras, aunque no serán tan precisos al no ser específicos de la cámara utilizada.

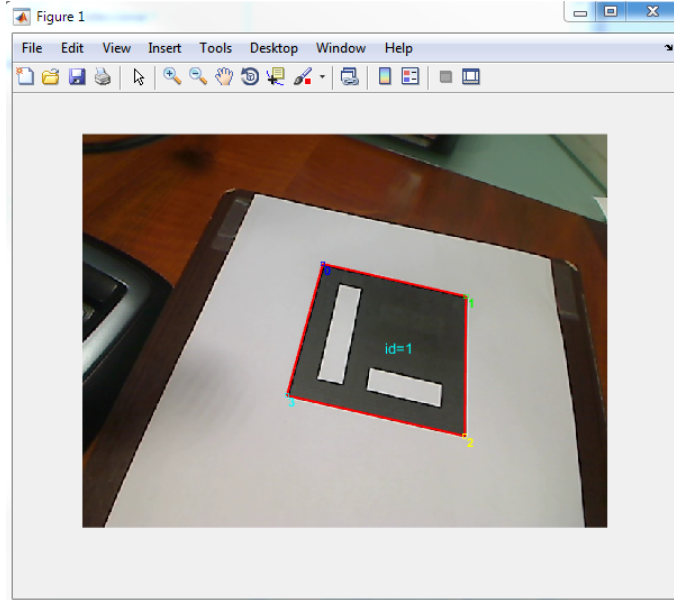
Los datos de calibración de la cámara y las coordenadas 3D de los marcadores se cargan con los ficheros suministrados:

- **calib_LogitechC300.m**: fichero con los datos de calibración de la cámara Logitech C300
- **calib_LogitechWP9000.m**: fichero con los datos de calibración de la cámara Logitech WP9000
- **calib_Trust.m**: fichero con los datos de calibración de la cámara Trust
- **board_data.m**: fichero con los datos del markersBoard.pdf
- **marker_data.m**: fichero con los datos del marker_id1.pdf

4.2. Detección de marcas y cálculo de la homografía del plano (Obligatorio).

Se utilizará el ejecutable suministrado que detecta y extrae las esquinas de las marcas. Se usará a través del script **markers.m** que ya está incluido en los ficheros **pose1.m** y **pose2.m**.

```
[id, points, im] = markers;
```



El programa lanzará la captura de la cámara y detección de marcas en tiempo real. Para devolver los resultados basta con pulsar las teclas **ESC** ó **Q**. A continuación (ya integrado) se muestran los resultados con el script suministrado **showmarkers.m**.

El alumno deberá completar el código de la función **homografia.m** para el cálculo de la homografía entre el plano de la marca y el plano de la imagen. En el fichero se indican los lugares donde se debe incluir el código y una descripción de los cálculos. A continuación se resumen las ecuaciones involucradas.

$$\text{Si } \tilde{M} \cong \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \in \Pi \Rightarrow \tilde{m} \cong A \cdot \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad \begin{bmatrix} \rho \cdot u \\ \rho \cdot v \\ \rho \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$(h_{31}X + h_{32}Y + h_{33}) \cdot u = h_{11}X + h_{12}Y + h_{13}$$

$$(h_{31}X + h_{32}Y + h_{33}) \cdot v = h_{21}X + h_{22}Y + h_{23}$$

$$\mathbf{A}_{xy} = \begin{bmatrix} -X^{(1)} & -Y^{(1)} & -1 & 0 & 0 & 0 & X^{(1)}u^{(1)} & Y^{(1)}u^{(1)} & u^{(1)} \\ 0 & 0 & 0 & -X^{(1)} & -Y^{(1)} & -1 & X^{(1)}v^{(1)} & Y^{(1)}v^{(1)} & v^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$$

$$\mathbf{A}_{xy} \cdot \mathbf{h} = 0$$

$$\min_{\mathbf{h}} [(\mathbf{A}_{xy} \mathbf{h})^2]$$

$$SVD(\mathbf{A}_{xy}) \rightarrow \mathbf{A}_{xy} = \mathbf{U}\mathbf{D}\mathbf{V}^T \Rightarrow \mathbf{h} = \mathbf{v}_9$$

Se debe mostrar la condición del sistema de ecuaciones lineales a rango 8 (ValSingular(1)/ ValSingular (8)).

Normalización de coordenadas:

Como tarea adicional se debe incorporar la transformación de normalización de coordenadas para obtener unos resultados más robustos en la estimación de **H**. Se deben definir dos transformaciones (**T**₁, **T**₂) para las coordenadas de imagen y las coordenadas del plano respectivamente. (Nota: el código de transformación ya está incluido, solo es preciso definir las matrices de transformación). Comparar los resultados en el valor de la condición del sistema de ecuaciones lineales y el efecto del ruido en la estimación de **H** con o sin normalización

Para elegir los valores de (**T**₁, **T**₂), se propone utilizar una transformación que haga que el centroide de todos los puntos sea el origen y la media de las distancia al origen sea $\sqrt{2}$. Por ejemplo:

$$T = \begin{bmatrix} 1/\sigma_x & 0 & -\mu_x/\sigma_x \\ 0 & 1/\sigma_y & -\mu_y/\sigma_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \tilde{\mathbf{m}} &= T_1 \cdot \mathbf{m} \\ \tilde{\mathbf{m}}_w &= T_2 \cdot \mathbf{m}_w \end{aligned} \quad \tilde{\mathbf{m}} \cong \hat{H} \cdot \tilde{\mathbf{m}}_w \quad H \cong T_1^{-1} \cdot \hat{H} \cdot T_2$$

4.3. Cálculo de la Rotación y Traslación entre cámara y marca.

Se deberán obtener la matriz de rotación (**R**) y el vector de traslación (**t**) a partir de la homografía **H** y la matriz de parámetros intrínsecos **A**. Se deberán modificar los ficheros **pose1.m** y **pose2.m** incluyendo el código necesario en los lugares indicados. Se indican a continuación las ecuaciones involucradas.

$$\begin{bmatrix} \bar{\mathbf{r}}_1 & \bar{\mathbf{r}}_2 & \bar{\mathbf{t}} \end{bmatrix} = \lambda \cdot A^{-1} \begin{bmatrix} \bar{\mathbf{h}}_1 & \bar{\mathbf{h}}_2 & \bar{\mathbf{h}}_3 \end{bmatrix} \quad \|\bar{\mathbf{r}}_1\| = 1 \quad \|\bar{\mathbf{r}}_2\| = 1$$

$$R^* = \begin{bmatrix} A^{-1}\bar{\mathbf{h}}_1 & A^{-1}\bar{\mathbf{h}}_2 & (A^{-1}\bar{\mathbf{h}}_1 \times A^{-1}\bar{\mathbf{h}}_2) \end{bmatrix}$$

$$SVD(R^*) \rightarrow R^* = U \cdot D \cdot V^T \quad R = U \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & |U \cdot V^T| \end{bmatrix} \cdot V^T$$

$$D = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \quad \lambda = 1/\sigma_1 \quad \bar{\mathbf{t}} = A^{-1}\bar{\mathbf{h}}_3 / \sigma_1$$

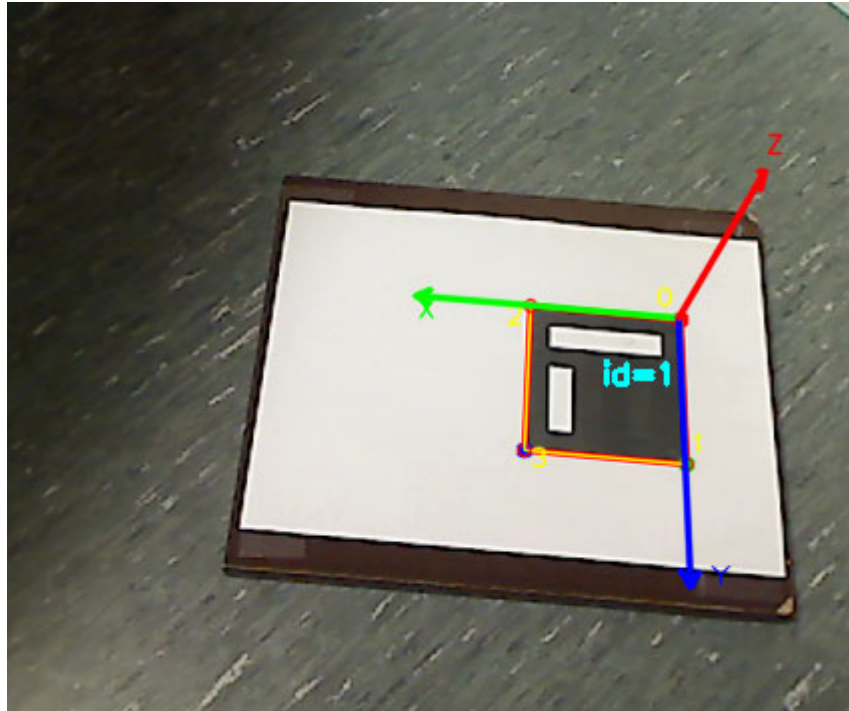
Para comprobar si que cálculo de R, t es correcto, el scriptp Malab (**pose1**, **pose2**) incorpora el cálculo y muestra el error de reproyección medio para las esquinas detectadas en los marcadores:

$$\begin{aligned}\widetilde{m}_i^r &\cong A[R \quad t]\widetilde{M}_i \\ m_i^r &= \widetilde{m}_i^r / \widetilde{m}_i^r(3) \\ e &= \frac{1}{n} \sum_{i=1}^n \left\| m_i - m_i^r \right\|^2\end{aligned}$$

4.4. Dibujar sistema de coordenadas 3D sobre la imagen (OPCIONAL)

Como tarea adicional se pide dibujar sobre la marca en la imagen el sistema de coordenadas 3D utilizando la ecuación de proyección para proyectar los tres ejes coordenados.

$$\tilde{m} \cong A[R \quad t]\tilde{M}$$



4.5. Estudio de la estabilidad frente al ruido

Se deben estudiar los resultados previos cuando añadimos ruido a los datos de coordenadas de imagen. Para ello se añadirá el siguiente código antes de calcular la homografía:

```
points = points + randn(size(points,1), size(points,2), size(points,3));
```


4.6. OPCIONAL: corrección distorsión de la imagen

Eliminar la distorsión de los puntos detectados de la imagen corrigiendo las coordenadas obtenidas por la función `markers`.

Se utilizarán los datos de calibración completos con la distorsión: `kc=[k1; k2; p1; p2; k3]`

Utilizar la función `comp_distortion oulu` incluida en de la toolbox de calibración para transformar los datos de la variable `points`.

```
[x] = comp_distortion oulu(xd, kc)
```

INPUT: `xd`: distorted (normalized) point coordinates in the image plane
(2xN matrix)

`kc`: Distortion coefficients (radial and tangential) (4x1 vector)

OUTPUT: `x`: undistorted (normalized) point coordinates in the image plane
(2xN matrix)

Nota:

- Esta función requiere que los datos de entrada estén en coordenadas normalizadas (se debe restar del punto principal y dividir por la focal).

$$q_x = (m_x - c_x) / f_x$$
$$q_y = (m_y - c_y) / f_y$$

- La función devuelve el resultado en coordenadas normalizadas por lo que se deberán transformar de nuevo a coordenadas en pixels.

$$m_x = q_x * f_x + c_x$$
$$m_y = q_y * f_y + c_y$$

- Se recomienda utilizar expresiones vectoriales compactas de Matlab usando los arrays: `points, fc, cc`

5. Documentos generados y entregados

La realización de la práctica debe realizarse de forma individual. Es necesario entregar **un documento en el que se describa el trabajo realizado** y que incluya los resultados relacionados a continuación, **así como los programas empleados en su obtención**. Se entregará un zip, en el que se indique en nombre del alumno y los escenarios utilizados. Por favor, se ruega no mandar las imágenes de entrada, para que el fichero no sea demasiado grande.

Los ficheros creados y modificados deberán incluir al menos los siguientes:

- **pose1.m**: solución para la práctica con el escenario 1
- **pose2.m** solución para la práctica con el escenario 2
- **homografia.m**: solución para cálculo de la homografía entre planos proyectivos
- **axis3D.m**: solución para el dibujo de los ejes de coordenadas 3D en la imagen (SC mundo) (Opcional)

Resultados:

1. Escenario 1 (una marca): **Condición sistema de ecuaciones, Homografía, R, t, error**
 - a) Sin normalización - sin ruido
 - b) Con normalización - sin ruido
 - c) Sin normalización - con ruido
 - d) Con normalización - ruido
2. Escenario 2 (seis marcas): **Condición sistema de ecuaciones, Homografía, R, t, error**
 - a) Sin normalización - sin ruido
 - b) Con normalización - sin ruido
 - c) Sin normalización - con ruido
 - d) Con normalización - ruido

OPCIONAL: Resultados incluyendo la corrección de distorsión