

# Realidad Aumentada

## Objetivos

- Aprender a calibrar una cámara usando OpenCV.
- Aprender a utilizar la calibración de la cámara para realizar Realidad Aumentada proyectando un modelo 3D sobre la imagen.
- Aprender a leer un flujo de video.
- Aprender a utilizar OpenCV para almacenar/recuperar parámetros de un fichero.

## Requerimientos mínimos (hasta un 7).

1. Utiliza la [código de ejemplo que ofrece OpenCV](#) para calibrar una cámara. Puedes utilizar tu propia cámara con el patrón de calibración proporcionado o usar un vídeo suministrado si no tienes cámara.
2. Crea el programa “augReal” que utiliza los parámetros de calibración obtenidos en el paso 1 para superponer información virtual 3d sobre la imagen capturada de la cámara o el vídeo proporcionado. La secuencia a realizar es de forma esquemática:

Para cada imagen del flujo de vídeo de entrada (utiliza [cv::VideoCapture](#) para leer de un flujo de vídeo), el programa debería realizar los siguientes pasos:

1. Detectar el tablero de calibración usando [cv::findChessboardCorners](#), y refinar las esquinas detectadas con [cv::cornerSubPix](#).
2. Estimar la orientación de la cámara respecto al patrón (su “pose”) usando [cv::solvePnP](#).
3. Proyectar sobre la imagen información 3D. Como mínimo se deberían dibujar los ejes XYZ en colores Rojo, Verde y Azul sobre el punto origen del sistemas de referencia 3D (el patrón). El tamaño de los ejes debería ser el mismo que el tamaño de los cuadrados del tablero. Utilizar [cv::projectPoints](#) para calcular las coordenadas de imagen al proyectar puntos 3D and [cv::line](#) para dibujar sobre la imagen las líneas que forman los ejes.

La CLI del programa debería ser:

```
augReal rows cols size intrinsics.yml <input video-file|cam-idx>
```

donde rows, cols and size dan la geometría del tablero de calibración, intrinsics.yml corresponde al fichero con los parámetros de calibración generado con la herramienta de OpenCV. Para leer los parámetros intrínsecos del fichero intrinsics.yml utiliza la clase [cv::FileStorage](#).

## Opcional (hasta 3 puntos)

Muestra una imagen “virtual” sobre el tablero visto el flujo de vídeo como si el tablero fuera esta imagen. Para ello debes calcular la transformación en perspectiva necesaria para proyectar el rectángulo externo en coordenadas de imagen que define el tablero al proyectarse sobre la imagen y las esquinas de la imagen virtual a dibujar, esto se calcula con [cv::getPerspectiveTransform](#) y después debes dibujar esta imagen “virtual” sobre la imagen del tablero con [cv::warpPerspective](#).

Opcionalmente también podrías, en vez de renderizar una imagen, renderizar un vídeo.

Añade un opción extra al programa augReal

```
[-i <img>] [-v <video>]
```

para indicar que queremos renderizar una imagen|vídeo sobre el tablero.