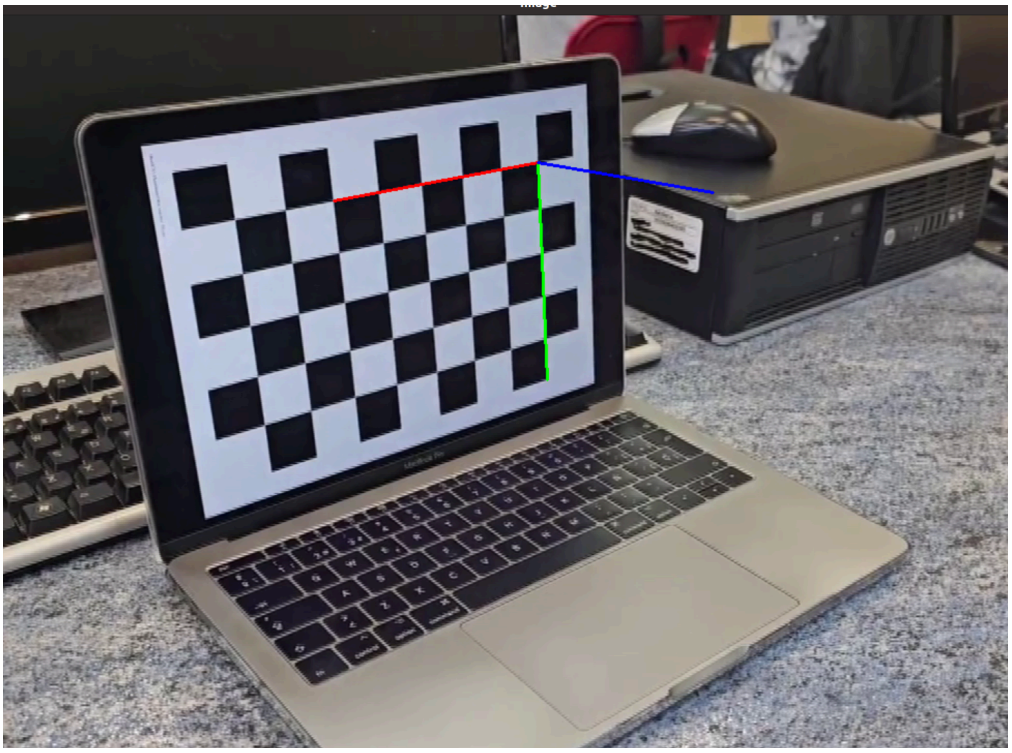
Camera Calibration and 3D information

Goals

Create a program that prints a 3D axis on this video showing the calibration pattern.



- Learning how to calibrate a camera using OpenCV.
- Learning how to use calibration to augment reality.

Minimum requirements (60%)

1. Calibrate the camera that recorded the video file provided using the images in this folder.
2. Create a program to augment reality using the file with the camera parameters. The idea is to create a program that reads the video recorded with the calibrated camera and superimposes information on it.

For each video frame, the program should proceed as follows:

1. Detect the board using cv::findChessboardCorners, and refine the corners with cv::cornerSubPix.
2. Estimate the camera pose with respect to the board using cv::solvePnP.
3. Drawing on the image is a simple 3D scene. The 3D axis is placed in the centre of the reference system in colours red (X-axis), green (Y-axis), and blue (Z-axis). The

size of the axis should be the same as the board squares. You can use the function cv::projectPoints and cv::line for that.

The program should be used as:

```
augReal size intrisics.yml videofile
```

The input parameters are:

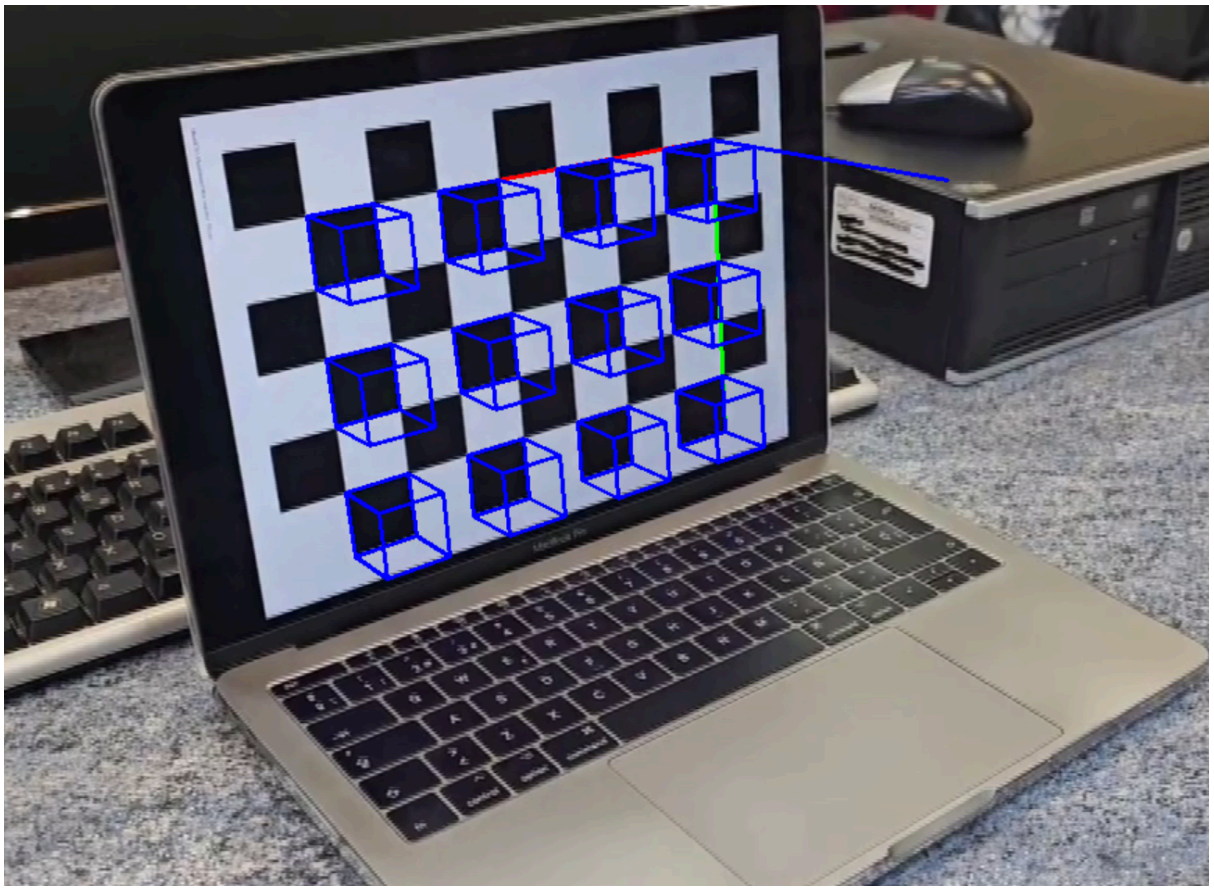
size: the size of the axis to be drawn

intrinsics.yml: the intrinsic parameters stored in the .yml file generated with the previous program

video-file: video sequence.

Optional requirements (40%)

Show a cube 1x1x1 on each black square of the calibration pattern.



Example reading video

```
#include <opencv2/highgui.hpp>
#include <iostream>
int main(int argc,char **argv){
   std::string pathToVideo=argv[1];
   cv::VideoCapture video(pathToVideo);

   cv::Mat image;
   while(video.grab()){

        video.retrieve(image);
        cv::imshow("image",image);
        cv::waitKey(10);
   }

}
```